THE NATA CODE - PROGRAMMER'S MANUAL

W. L. Bade and J. M. Yos
AVCO SYSTEMS DIVISION
201 Lowell Street
Wilmington, Massachusetts 01887

NAS9-9744
CR-141744
2 September 1975
Final Report
Volume III

(NASA-CF-141744) THE NATA CODE: THEORY AND
ANALYSIS. VOLUME 3: PROGRAMMER'S MANUAL
Final Report (Avco Systems Div., Wilmington,
Mass.) 635 p HC $16.25

Prepared for
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas 77058
The NATA code is a computer program for calculating quasi-one-dimensional gas flow in axisymmetric and two-dimensional nozzles and rectangular channels. The code also computes stagnation point conditions on axisymmetric or two-dimensional models and the conditions on blunt wedges inserted into the flow. The theory and analysis underlying the calculations have been documented in Volume I of this final report. Code inputs, outputs, and precoded data have been defined in Volume II, which is a user's manual. The present, third volume of the final report is a programmer's manual for the code. It provides a listing of the Fortran IV source program; a complete directory of Fortran symbols; a discussion of the purpose and method of operation of each subroutine (including mathematical analyses of special algorithms); and a discussion of the operation of the code on IBM/360 and UNIVAC 1108 systems, including required control cards and the overlay structure used to accommodate the code to the limited core size of the 1108. In addition, this volume provides similar information to document the programming of the NOZFIT code, which is employed to set up nozzle profile curvefits for use in NATA.
PREFACE

This is the third volume of a final report documenting the development of a computer program (NATA) for calculating the flow in arc-heated wind tunnels and the conditions on models tested in such reentry simulation facilities. Volume I of the report provides a general summary of the code's capabilities and presents the theory and analysis underlying its operation. Volume II is a user's manual which defines NATA's inputs, outputs, and precode'd data on standard gas models and nozzle geometries. The present volume is a programmer's manual for NATA. It documents the programming of NATA in sufficient detail to support code modifications, and provides information needed for running the code on the UNIVAC 1108 under the EXEC II system, or on the IBM 360/75 under HASP II.
# TABLE OF CONTENTS

1. Introduction ............................................. 1

2. Structure and Operation ................................. 2
   2.1 Brief Description of the NATA Code .................. 2
   2.2 Subroutines and Linkage .............................. 2
   2.3 UNIVAC 1108 Version .................................. 12
   2.4 IBM 360 Version ....................................... 16

3. Analysis of Routines ..................................... 20
   3.1 Main Program .......................................... 20
   3.2 Subroutine AESØLN .................................... 28
   3.3 Subroutine AGSØLN .................................... 30
   3.4 Subroutine AXFIT ...................................... 32
   3.5 Subroutine AXSECT .................................... 33
   3.6 Subroutine BLAYER .................................... 33
   3.7 Subroutine BLCALL .................................... 44
   3.8 Subroutine BXSECT .................................... 45
   3.9 Subroutine CØMM ...................................... 46
   3.10 Subroutine CXSECT .................................... 61
   3.11 Subroutine DERIVS .................................... 61
   3.12 Subroutine DSMSØL .................................... 62
   3.13 Subroutine DUMP ...................................... 64
   3.14 Subroutine DUMPEX .................................... 64
   3.15 Subroutine ELCØND .................................... 65
   3.16 Subroutine ELTIME .................................... 65
   3.17 Subroutine EPART ..................................... 66
   3.18 Subroutine EQCALC .................................... 67
   3.19 Subroutine EXACT ..................................... 70
   3.20 Function EXP .......................................... 73
   3.21 Subroutine FINDX ..................................... 76
   3.22 Subroutine FRØZEQ .................................... 77
   3.23 Subroutine GEØM ...................................... 91
   3.24 Subroutine GEØMAR .................................... 101
   3.25 Subroutine INGAS ..................................... 103
   3.26 Subroutine INIT ....................................... 105
   3.27 Subroutine KANDMU .................................... 105
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>KINT</td>
<td>106</td>
</tr>
<tr>
<td>LIST</td>
<td>107</td>
</tr>
<tr>
<td>MATINV</td>
<td>108</td>
</tr>
<tr>
<td>MODEL</td>
<td>108</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>115</td>
</tr>
<tr>
<td>NEXTMP</td>
<td>120</td>
</tr>
<tr>
<td>N0NEXT</td>
<td>121</td>
</tr>
<tr>
<td>NRMAX</td>
<td>139</td>
</tr>
<tr>
<td>OUT</td>
<td>139</td>
</tr>
<tr>
<td>OUT1</td>
<td>140</td>
</tr>
<tr>
<td>PERT</td>
<td>148</td>
</tr>
<tr>
<td>PI0MEG</td>
<td>149</td>
</tr>
<tr>
<td>PR0P</td>
<td>150</td>
</tr>
<tr>
<td>PRTA</td>
<td>150</td>
</tr>
<tr>
<td>PUTQIN</td>
<td>151</td>
</tr>
<tr>
<td>QC0UL</td>
<td>154</td>
</tr>
<tr>
<td>QEX</td>
<td>155</td>
</tr>
<tr>
<td>QEXP</td>
<td>156</td>
</tr>
<tr>
<td>QINTRP</td>
<td>156</td>
</tr>
<tr>
<td>Q1J</td>
<td>157</td>
</tr>
<tr>
<td>QMI0X</td>
<td>157</td>
</tr>
<tr>
<td>QREPP</td>
<td>158</td>
</tr>
<tr>
<td>QSAME</td>
<td>159</td>
</tr>
<tr>
<td>QTAB</td>
<td>159</td>
</tr>
<tr>
<td>Q11</td>
<td>159</td>
</tr>
<tr>
<td>Q12</td>
<td>159</td>
</tr>
<tr>
<td>Q13</td>
<td>160</td>
</tr>
<tr>
<td>Q14</td>
<td>160</td>
</tr>
<tr>
<td>RADIUS</td>
<td>160</td>
</tr>
<tr>
<td>READ</td>
<td>161</td>
</tr>
<tr>
<td>RESET</td>
<td>166</td>
</tr>
<tr>
<td>RESTMP</td>
<td>166</td>
</tr>
<tr>
<td>RNKT</td>
<td>174</td>
</tr>
<tr>
<td>SH0CK</td>
<td>178</td>
</tr>
<tr>
<td>SIMQ</td>
<td>181</td>
</tr>
<tr>
<td>STUNL</td>
<td>181</td>
</tr>
<tr>
<td>THERM</td>
<td>186</td>
</tr>
<tr>
<td>TABLE OF CONTENTS</td>
<td>Page</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>(Continued)</td>
<td></td>
</tr>
<tr>
<td>3.65 Subroutine THRT</td>
<td>188</td>
</tr>
<tr>
<td>3.66 Subroutine TRANS</td>
<td>188</td>
</tr>
<tr>
<td>3.67 Subroutine WEDGE</td>
<td>192</td>
</tr>
<tr>
<td>3.68 Subroutine WESLN</td>
<td>193</td>
</tr>
<tr>
<td>3.69 Subroutine XSECT</td>
<td>195</td>
</tr>
<tr>
<td>4. Glossary of Fortran Symbols</td>
<td>212</td>
</tr>
<tr>
<td>4.1 Unlabelled Common</td>
<td></td>
</tr>
<tr>
<td>4.2 Common /AECEOM/</td>
<td>224</td>
</tr>
<tr>
<td>4.3 Common /AREA/</td>
<td>224</td>
</tr>
<tr>
<td>4.4 Common /AVG/</td>
<td>225</td>
</tr>
<tr>
<td>4.5 Common /BL/</td>
<td>225</td>
</tr>
<tr>
<td>4.6 Common /BLNE/</td>
<td>226</td>
</tr>
<tr>
<td>4.7 Common /BLNUT/</td>
<td>227</td>
</tr>
<tr>
<td>4.8 Common /BLRAD/</td>
<td>227</td>
</tr>
<tr>
<td>4.9 Common /CERROR/</td>
<td>227</td>
</tr>
<tr>
<td>4.10 Common /CHAN/</td>
<td>227</td>
</tr>
<tr>
<td>4.11 Common /COLDSP/</td>
<td>228</td>
</tr>
<tr>
<td>4.12 Common /CONEVRT/</td>
<td>228</td>
</tr>
<tr>
<td>4.13 Common /DRV/</td>
<td>229</td>
</tr>
<tr>
<td>4.14 Common /BELEM/</td>
<td>229</td>
</tr>
<tr>
<td>4.15 Common /ELEM/</td>
<td>230</td>
</tr>
<tr>
<td>4.16 Common /EQC/</td>
<td>230</td>
</tr>
<tr>
<td>4.17 Common /EQC2/</td>
<td>230</td>
</tr>
<tr>
<td>4.18 Common /ERROR/</td>
<td>231</td>
</tr>
<tr>
<td>4.19 Common /GLIM2/</td>
<td>231</td>
</tr>
<tr>
<td>4.20 Common /INGNE/</td>
<td>231</td>
</tr>
<tr>
<td>4.21 Common /LN/</td>
<td>231</td>
</tr>
<tr>
<td>4.22 Common /MASSFL/</td>
<td>231</td>
</tr>
<tr>
<td>4.23 Common /MDT/</td>
<td>232</td>
</tr>
<tr>
<td>4.24 Common /MODEAR/</td>
<td>233</td>
</tr>
<tr>
<td>4.25 Common /MODEP2/</td>
<td>234</td>
</tr>
<tr>
<td>4.26 Common /MODEP2/</td>
<td>234</td>
</tr>
<tr>
<td>4.27 Common /NEQ/</td>
<td>235</td>
</tr>
<tr>
<td>4.28 Common /NEWMP/</td>
<td>235</td>
</tr>
<tr>
<td>4.29 Common /NQZ2/</td>
<td>235</td>
</tr>
<tr>
<td>4.30 Common /OUTMRD/</td>
<td>236</td>
</tr>
<tr>
<td>Page</td>
<td>TABLE OF CONTENTS (Continued)</td>
</tr>
<tr>
<td>------</td>
<td>--------------------------------</td>
</tr>
<tr>
<td>237</td>
<td>Common /\OUTPUT/</td>
</tr>
<tr>
<td>239</td>
<td>Common /\POLYAT/</td>
</tr>
<tr>
<td>239</td>
<td>Common /\RDLIST/</td>
</tr>
<tr>
<td>239</td>
<td>Common /\RDMAIN/</td>
</tr>
<tr>
<td>240</td>
<td>Common /\RDM\ØD/</td>
</tr>
<tr>
<td>240</td>
<td>Common /\RD\ØUT/</td>
</tr>
<tr>
<td>240</td>
<td>Common /\RDTR/</td>
</tr>
<tr>
<td>240</td>
<td>Common /\RDNEDG/</td>
</tr>
<tr>
<td>241</td>
<td>Common /\REAC/</td>
</tr>
<tr>
<td>241</td>
<td>Common /\READAT/</td>
</tr>
<tr>
<td>242</td>
<td>Common /\RENE/</td>
</tr>
<tr>
<td>245</td>
<td>Common /\RESPRP/</td>
</tr>
<tr>
<td>246</td>
<td>Common /\SPEC/</td>
</tr>
<tr>
<td>247</td>
<td>Common /\SS/</td>
</tr>
<tr>
<td>247</td>
<td>Common /\STAG/</td>
</tr>
<tr>
<td>247</td>
<td>Common /\STEPS/</td>
</tr>
<tr>
<td>247</td>
<td>Common /\SWITCH/</td>
</tr>
<tr>
<td>248</td>
<td>Common /\TAP\ØUT/</td>
</tr>
<tr>
<td>248</td>
<td>Common /\TEMPRY/</td>
</tr>
<tr>
<td>248</td>
<td>Common /\THRT/</td>
</tr>
<tr>
<td>249</td>
<td>Common /\TNCE/</td>
</tr>
<tr>
<td>249</td>
<td>Common /\TNDC/</td>
</tr>
<tr>
<td>249</td>
<td>Common /\TNE/</td>
</tr>
<tr>
<td>250</td>
<td>Common /\TNEQ/</td>
</tr>
<tr>
<td>251</td>
<td>Common /\TNERK/</td>
</tr>
<tr>
<td>253</td>
<td>Common /\TN\ØNEQ/</td>
</tr>
<tr>
<td>254</td>
<td>Common /\TRANS1/</td>
</tr>
<tr>
<td>254</td>
<td>Common /\TRANS2/</td>
</tr>
<tr>
<td>254</td>
<td>Common /\TRANS3/</td>
</tr>
<tr>
<td>254</td>
<td>Common /\TRANS4/</td>
</tr>
<tr>
<td>255</td>
<td>Common /\TRANS5/</td>
</tr>
<tr>
<td>255</td>
<td>Common /\TRANS6/</td>
</tr>
<tr>
<td>255</td>
<td>Common /\TRANS7/</td>
</tr>
<tr>
<td>257</td>
<td>Common /\TRANS8/</td>
</tr>
<tr>
<td>257</td>
<td>Common /\TRANS9/</td>
</tr>
<tr>
<td>258</td>
<td>Common /\TRPR\ØP/</td>
</tr>
<tr>
<td>258</td>
<td>Common /\WM\ØD/</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS
(Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.69</td>
<td>Main Program</td>
<td>258</td>
</tr>
<tr>
<td>4.70</td>
<td>Subroutine AESØLN</td>
<td>260</td>
</tr>
<tr>
<td>4.71</td>
<td>Subroutine AGSØLN</td>
<td>261</td>
</tr>
<tr>
<td>4.72</td>
<td>Subroutine AXFIT</td>
<td>262</td>
</tr>
<tr>
<td>4.73</td>
<td>Subroutine AXSECT</td>
<td>262</td>
</tr>
<tr>
<td>4.74</td>
<td>Subroutine BLAYER</td>
<td>263</td>
</tr>
<tr>
<td>4.75</td>
<td>Subroutine BLCALL</td>
<td>270</td>
</tr>
<tr>
<td>4.76</td>
<td>Subroutine BXSECT</td>
<td>270</td>
</tr>
<tr>
<td>4.77</td>
<td>Subroutine CØMM</td>
<td>270</td>
</tr>
<tr>
<td>4.78</td>
<td>Subroutine CXSECT</td>
<td>274</td>
</tr>
<tr>
<td>4.79</td>
<td>Subroutine DERIVS</td>
<td>279</td>
</tr>
<tr>
<td>4.80</td>
<td>Subroutine DSMØL</td>
<td>279</td>
</tr>
<tr>
<td>4.81</td>
<td>Subroutine DUMP</td>
<td>276</td>
</tr>
<tr>
<td>4.82</td>
<td>Subroutine DUMPEx</td>
<td>276</td>
</tr>
<tr>
<td>4.83</td>
<td>Subroutine ELCØND</td>
<td>276</td>
</tr>
<tr>
<td>4.84</td>
<td>Subroutine ELTIME</td>
<td>276</td>
</tr>
<tr>
<td>4.85</td>
<td>Subroutine EPART</td>
<td>277</td>
</tr>
<tr>
<td>4.86</td>
<td>Subroutine EQCALC</td>
<td>277</td>
</tr>
<tr>
<td>4.87</td>
<td>Subroutine EXACT</td>
<td>278</td>
</tr>
<tr>
<td>4.88</td>
<td>Function EXP</td>
<td>279</td>
</tr>
<tr>
<td>4.89</td>
<td>Subroutine FINDX</td>
<td>279</td>
</tr>
<tr>
<td>4.90</td>
<td>Subroutine FRMSEQ</td>
<td>280</td>
</tr>
<tr>
<td>4.91</td>
<td>Subroutine GEMM</td>
<td>284</td>
</tr>
<tr>
<td>4.92</td>
<td>Subroutine GEMMAR</td>
<td>285</td>
</tr>
<tr>
<td>4.93</td>
<td>Subroutine INGAS</td>
<td>286</td>
</tr>
<tr>
<td>4.94</td>
<td>Subroutine INIT</td>
<td>287</td>
</tr>
<tr>
<td>4.95</td>
<td>Subroutine KANDMU</td>
<td>287</td>
</tr>
<tr>
<td>4.96</td>
<td>Subroutine KINT</td>
<td>288</td>
</tr>
<tr>
<td>4.97</td>
<td>Subroutine LIST</td>
<td>289</td>
</tr>
<tr>
<td>4.98</td>
<td>Subroutine MATINV</td>
<td>290</td>
</tr>
<tr>
<td>4.99</td>
<td>Subroutine MØDEL</td>
<td>291</td>
</tr>
<tr>
<td>4.100</td>
<td>Subroutine NEWRAP</td>
<td>295</td>
</tr>
<tr>
<td>4.101</td>
<td>Subroutine NEXTMP</td>
<td>296</td>
</tr>
<tr>
<td>4.102</td>
<td>Subroutine NØNEQ</td>
<td>296</td>
</tr>
<tr>
<td>4.103</td>
<td>Subroutine NRMAX</td>
<td>301</td>
</tr>
<tr>
<td>4.104</td>
<td>Subroutine ØUTI</td>
<td>301</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.105 Subroutine PERT</td>
<td>304</td>
</tr>
<tr>
<td>4.106 Function PIØMEG</td>
<td>305</td>
</tr>
<tr>
<td>4.107 Subroutine PRØP</td>
<td>305</td>
</tr>
<tr>
<td>4.108 Subroutine PRTA</td>
<td>305</td>
</tr>
<tr>
<td>4.109 Subroutine PUTQIN</td>
<td>305</td>
</tr>
<tr>
<td>4.110 Subroutine QCØUL</td>
<td>306</td>
</tr>
<tr>
<td>4.111 Subroutine QEX</td>
<td>306</td>
</tr>
<tr>
<td>4.112 Subroutine QEXP</td>
<td>307</td>
</tr>
<tr>
<td>4.113 Subroutine QINTRP</td>
<td>308</td>
</tr>
<tr>
<td>4.114 Subroutine QLJ</td>
<td>308</td>
</tr>
<tr>
<td>4.115 Subroutine QMTX</td>
<td>309</td>
</tr>
<tr>
<td>4.116 Subroutine QREPP</td>
<td>309</td>
</tr>
<tr>
<td>4.117 Subroutine QSAME</td>
<td>309</td>
</tr>
<tr>
<td>4.118 Subroutine QTAB</td>
<td>310</td>
</tr>
<tr>
<td>4.119 Subroutine Q11</td>
<td>310</td>
</tr>
<tr>
<td>4.120 Subroutine Q12</td>
<td>311</td>
</tr>
<tr>
<td>4.121 Subroutine Q13</td>
<td>311</td>
</tr>
<tr>
<td>4.122 Subroutine Q14</td>
<td>312</td>
</tr>
<tr>
<td>4.123 Subroutine RADIUS</td>
<td>312</td>
</tr>
<tr>
<td>4.124 Subroutine READ</td>
<td>313</td>
</tr>
<tr>
<td>4.125 Subroutine RESET</td>
<td>317</td>
</tr>
<tr>
<td>4.126 Subroutine RESTMP</td>
<td>317</td>
</tr>
<tr>
<td>4.127 Subroutine RNKT</td>
<td>319</td>
</tr>
<tr>
<td>4.128 Subroutine SHØCK</td>
<td>322</td>
</tr>
<tr>
<td>4.129 Subroutine SIMQ</td>
<td>324</td>
</tr>
<tr>
<td>4.130 Subroutine STUNTS</td>
<td>325</td>
</tr>
<tr>
<td>4.131 Subroutine THERM</td>
<td>327</td>
</tr>
<tr>
<td>4.132 Subroutine THRØAT</td>
<td>329</td>
</tr>
<tr>
<td>4.133 Subroutine TRANSP</td>
<td>329</td>
</tr>
<tr>
<td>4.134 Subroutine WEDGE</td>
<td>330</td>
</tr>
<tr>
<td>4.135 Subroutine WESØLN</td>
<td>337</td>
</tr>
<tr>
<td>4.136 Subroutine XSECT</td>
<td>338</td>
</tr>
</tbody>
</table>

5. References to Common Variables ........................................... 345

6. Listing of the DATA Code .................................................. 374
# TABLE OF CONTENTS
(Continued)

<table>
<thead>
<tr>
<th>Appendix - Programmer's Manual for the NOZFIT Code</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.1 Analysis of NOZFIT Routines</td>
<td>576</td>
</tr>
<tr>
<td>A.2 Glossary of Fortran Symbols for NOZFIT</td>
<td>594</td>
</tr>
<tr>
<td>A.3 Listing of NOZFIT</td>
<td>608</td>
</tr>
<tr>
<td>References</td>
<td>624</td>
</tr>
</tbody>
</table>
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Illustration</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Overlay Structure for UNIVAC 1108 Version of NATA</td>
<td>14</td>
</tr>
<tr>
<td>2. Control Cards for Generating a NATA PCF Tape on the UNIVAC 1108</td>
<td>15</td>
</tr>
<tr>
<td>3. Control Cards for Running NATA on the UNIVAC 1108</td>
<td>15</td>
</tr>
<tr>
<td>4. Control Cards for Running NATA on the IBM 360</td>
<td>19</td>
</tr>
<tr>
<td>5. Flowchart of NATA Main Program</td>
<td>21</td>
</tr>
<tr>
<td>6. Flowchart of Subroutine BLAYER</td>
<td>35</td>
</tr>
<tr>
<td>7. Flowchart of Main Loop Over Reactions in Subroutine CΦMM</td>
<td>50</td>
</tr>
<tr>
<td>8. Flowchart of the Solution for the Derivatives of the Flow Variables in the Numerical Integration</td>
<td>75</td>
</tr>
<tr>
<td>9. Flowchart for the Determination of Sonic Conditions</td>
<td>79</td>
</tr>
<tr>
<td>10. Three Cases in the Algorithm for Finding the Throat Temperature</td>
<td>81</td>
</tr>
<tr>
<td>11. Flowchart of Subroutine FRΩEQ</td>
<td>83</td>
</tr>
<tr>
<td>12. Flowchart of Subroutine GEΩM</td>
<td>93</td>
</tr>
<tr>
<td>13. Approximations to the Density-Area Ratio Relation</td>
<td>95</td>
</tr>
<tr>
<td>14. Overall Flowchart of Subroutine MØDEL</td>
<td>110</td>
</tr>
<tr>
<td>15. Flowchart of the Normal Shock Solution</td>
<td>113</td>
</tr>
<tr>
<td>16. Flowchart of the Solution for Conditions at the Inviscid Stagnation Point</td>
<td>116</td>
</tr>
<tr>
<td>17. Flowchart of Subroutine NØNEQ</td>
<td>122</td>
</tr>
<tr>
<td>Illustration Description</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>18. Flowchart of Initialization Section of Subroutine N0NEQ</td>
<td>124</td>
</tr>
<tr>
<td>19. Flowchart of Switch to Downstream Region in Subroutine N0NEQ</td>
<td>126</td>
</tr>
<tr>
<td>20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine N0NEQ</td>
<td>127</td>
</tr>
<tr>
<td>21. Flowchart of Perturbation Solution and Switch to Integration in Subroutine N0NEQ</td>
<td>129</td>
</tr>
<tr>
<td>22. Flowchart of Controls for the Nonequilibrium Integration in Subroutine N0NEQ</td>
<td>134</td>
</tr>
<tr>
<td>23. Overall Flowchart of Subroutine OUT1</td>
<td>141</td>
</tr>
<tr>
<td>24. Flowchart of the Operations Produced by Calling Entry OUT2 of Subroutine OUT1</td>
<td>143</td>
</tr>
<tr>
<td>25. Iteration Scheme of Third Option for Specifying Reservoir Conditions</td>
<td>172</td>
</tr>
<tr>
<td>26. Flowchart of Subroutine RNKT</td>
<td>175</td>
</tr>
<tr>
<td>27. Schematic Flowchart of Subroutine XSECT</td>
<td>197</td>
</tr>
<tr>
<td>28. Geometry of the Condition IC0ND = 1</td>
<td>578</td>
</tr>
<tr>
<td>29. Geometry of the Condition IC0ND = 2</td>
<td>580</td>
</tr>
<tr>
<td>30. Geometry of the Condition IC0ND = 3</td>
<td>582</td>
</tr>
<tr>
<td>31. Geometry of the Condition IC0ND = 4</td>
<td>584</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

| I. | Derivatives Determined by EXACT During the Numerical Integration | 74 |
| II. | Subroutines Used in Cross Section Computations | 153 |
| III. | Indices Used for Calculation of Effective Coulomb Cross Sections in Subroutine XSECT | 209 |
The NATA code is a computer program for solving problems of steady, quasi-one-dimensional gas flow in nozzles. The code's capabilities, and the theory and analysis underlying its operation, have been documented in the first volume of this report (ref. 1). The inputs, outputs, and precoded data on standard gas models and nozzle geometries have been defined in the second volume (ref. 2). The present, third volume provides information required for running NATA on UNIVAC 1108 and IBM 360 computers, and documents the programming of NATA so that programmers and computer-oriented users will be able to evaluate the techniques used, maintain the code in the face of operating-system changes, convert it for use on other types of computer, introduce modifications and correct errors.

Section 2 provides a general description of NATA, including a catalog of its subroutines with a statement of function and a list of calling routines for each. This section also gives sample control-card decks for running the code on the UNIVAC 1108 under the EXEC II system, and on the IBM 360/75 under HASP II and CS 21.

Section 3 discusses each subroutine, explaining its purpose in the code and its method of operation. Many special algorithms are analyzed. Where appropriate, flowcharts are used to display the logic and clarify the methods used. The Fortran variables stored in common blocks and used in the individual routines are all defined in a glossary of symbols (Section 4). Section 5 lists the subroutines in which each common variable is set, used, or referenced as a subroutine argument. Finally, a complete listing of the source program is given in Section 6.
An appendix provides similar documentation of the NOZFIT code, a much smaller program provided for setting up nozzle profile curvefits for use in NATA. A user's manual for NOZFIT is included in Volume II (ref. 2). The appendix includes documentation of a general-purpose plotting subroutine (GRAPH) for producing plots on S-C 4020, S-C 4060, or other plotting equipment using the NAA software package for the SC-4020 (ref. 3).
2. STRUCTURE AND OPERATION

2.1 Brief Description of the NATA Code

NATA is a Fortran IV program consisting of a main program, two block data routines, and 66 subroutines. The source deck contains approximately 8500 cards. The program exists in two versions, one for use on the IBM 360 system, the other for the UNIVAC 1108. The IBM version is entirely in double precision, whereas the UNIVAC version is a single precision program with some double precision arrays and subroutines. One version can be converted into the other by inserting or removing most of the IMPLICIT REAL*8 (A-H, J-Z) cards which type all floating-point variables as double precision in the IBM 360 version.*

The IBM 360 version requires about 410K bytes of core storage, including buffers. The UNIVAC version is run on the 1108 using overlay, and requires about 123000 or 42500 words of storage. It fits into the two-bank processors at NASA/JSC with about 2000 words to spare.

2.2 Subroutines and Linkage

A list of the routines of NATA is presented below. Names of entry points are included, and are enclosed in parentheses to distinguish them from subroutine and function names. The function performed by each routine or entry point is described briefly, and in each case a complete list of calling routines is included. The order of the list is as follows: main program; block data routines; subroutines, functions, and entry points in alphanumeric order.

*Also, the IBM version contains two extra subroutines, EXP and RESET, for reasons explained below.
| **MAIN** | Main program; calls other routines to perform the required calculations. |
| **BLKD1** | BLOCK DATA routine containing precoded data on elements, species, reactions, gas models, and nozzle geometries. |
| **BLKD2** | BLOCK DATA routine containing precoded data on transport-property cross sections. |
| **AES\_LN** | Calculates the effective area ratio $A_e$ and its derivative $dA_e/dx$ from data on nozzle geometry and the boundary layer displacement thickness.* Called by GE\_M. |
| **AGS\_LN** | Calculates the geometric area ratio $A_g$ and the axial coordinate $x$ from data on the effective area ratio $A_e$ and the boundary layer displacement thickness. Called by FR\_ZEN, EQUIL, and AXFIT. |
| **AXFIT** | Calculates the axial coordinate $x$ corresponding to a given value of the effective area ratio $A_e$ in the nonequilibrium solution by the perturbation method. Called by N\_NEQ. |
| **AXSECT** | For a given list of species pairs, adds those pairs whose cross sections have not previously been specified to a particular step of the transport cross section calculation. Called by XSECT. |
| **BLAYER** | Calculates boundary layer momentum and displacement thicknesses, heat flux, and shear stress. Called by FR\_ZEN, EQUIL, and BICALL. |
| **BICALL** | Sets up and executes calls to BLAYER in the nonequilibrium solution. Called by DERIVS and PRTA. |

*Throughout this report, the mathematical notation is as defined in Volume I (ref. 1).*
BXSECT searches a parameter array for references to the indices of species pairs used in the cross section calculation; also determines the correspondence between entries in the parameter array and the steps in the cross section calculation. Called by XSECT.

C4OMM computes the species production terms $P_i \chi_i$ in the nonequilibrium solution; in runs with an electronic nonequilibrium gas model, also computes the radiative loss and the energy transfer to the electron gas. Called by DERIVS.

CXSECT determines the correspondence between species pairs in the master list of species and those in the species list for the current case. Called by XSECT.

DERIVS organizes the calculations of rates of change of the species concentrations and other flow variables in the nonequilibrium solution. Called by NØNEQ and RNKT.

DSMSØL performs simultaneous solution of a system of linear, inhomogeneous equations by calling subroutine SIMQ. Called by EQCALC, EXACT, NEWRAP, NØNEQ, and PERT.

DUMP prints name of routine in which an error has been detected and sets an error indicator to terminate the case. Called by AGSØLN, DSMSØL, EQCALC, FINDX, GEØM, GEØMAR, MATINV, NEWRAP, NØNEQ, PRØP, RESTMP, and WEDGE.

DUMPE prints a large diagnostic dump of common data when a case is terminated by a call to DUMP because of an error. Called by MAIN.

ELCØND computes the electrical conductivity of the gas mixture. Called by TRANSP.
ELTIME
Prints elapsed time since the beginning of the run and time since the last previous time message. Called by MAIN and MODEL.

EPART
Computes parameters for radiative energy loss and energy transfer to the electrons for a specified reaction. Called by COMM.

EQCALC
Computes mole fractions and thermodynamic conditions for thermochemical equilibrium at specified temperature and pressure. Called by INTA and MODEL.

(EQUIL)
Enter point in subroutine FRZEQ; organizes the calculation of the equilibrium flow solution. Called by MAIN.

EXACT
Computes the derivatives of the species concentrations and other flow variables in the non-equilibrium solution. Called by DERIVS.

EXP
In the IBM 360 version, the double precision exponential function $e^x$ with underflow suppression; included to avoid underflow messages. Not used in UNIVAC version. Called by BLAYER, COMM, EQCALC, NEWRAP, OUTL, PROP, RNKT, THERM, and WEDGE.

FINDX
Solves for the axial coordinate $x$ corresponding to a given geometric area ratio $A_g$ in the upstream or downstream region. Called by AGSOLN, AXFIT, FRZEQ, MAIN, and NEXTMP.

(FINDXC)
Enter point in subroutine FINDX; solves for the axial coordinate $x$ corresponding to a given half width of one of the profiles in a channel. Called by NEXTMP.

(FROZEN)
Enter point in subroutine FRZEQ; organizes the calculation of the frozen flow solution. Called by MAIN.
FRZSEQ Organizes the frozen and equilibrium flow solutions. The subroutine name is not referenced in NATA; called only through the entry points EQUIL and FRZEN.

GEOM Organizes the geometry calculations for the non-equilibrium solution. In the direct integration, it computes $A_e$ and $d\ln A_e/dx$; in the inverse method, it computes $\rho$ and $d\ln \rho/dx$. Called by COMM.

GEMMAR Computes the geometric area ratio $A_G$ and its derivative $d A_G/dx$ at a given axial coordinate $x$. Called by AGSOLN, BLAYER, FINDX, GEOM, OUT1, and THRALT.

(GMAR) Entry point of subroutine GEMMAR; computes the profile ordinate $y$ at a specified axial coordinate $x$. Called by COMM, OUT1, and RADIUS.

(GMAR2) Entry point of subroutine GEMMAR; computes the profile ordinates $y, z$ at a specified axial coordinate $x$ in a channel.Called by AGSOLN, COMM, OUT1, and RADIUS.

(GMAR3) Entry point of subroutine GEMMAR; computes the profile ordinates, $y, z$ and their derivatives $dy/dx, dz/dx$ at a specified axial coordinate $x$ in a channel. Called by AGSOLN, BLAYER, and FINDX.

INGAS Sets up arrays used in the chemical description of the gas mixture. Called by MAIN.

INIT Initializes same control parameters and nondimensionalizes the species thermochemical data using the reservoir temperature. Called by MAIN and RESTMP.
(INTA) Entry point of subroutine INGAS; organizes the calculation of equilibrium conditions in the upstream reservoir for given reservoir temperature and pressure. Called by MAIN and RESTMP.

KANDMU Calculates quantities proportional to the viscosity and translational thermal conductivity of the gas mixture. Called by TRANS.

KINT Computes quantities used in the transport property calculations. Called by TRANS.

LIST Prints the "problem summary" portion of the output for each case. Called by MAIN.

MATINV Computes the inverse $A_{kl}$ of the matrix $\alpha_k^i$ specifying the elemental composition of the independent species. Called by INGAS.

MODEL Computes stagnation conditions on models inserted into the flow. Called by FRZEQ, NNEQ, and PRTA.

NEWRAP Computes the equilibrium flow conditions at specified temperature and entropy. Called by FRZEQ, MAIN, NNEQ, and NMAX.

NEXTMP Determines the axial coordinate of the next location at which model condition calculations are to be done. Called by FRZEQ and NNEQ.

NNEQ Organizes the nonequilibrium flow solution. Called by MAIN.

NMAX Calculates the temperature and other conditions at the sonic point in an equilibrium flow starting at specified upstream reservoir conditions. Called by MAIN and RESTMP.

OUT Prints a list of definitions of output variables. Called by MAIN.
ØUTl
Prints conditions in the upstream reservoir and at the sonic points of the frozen and equilibrium solutions. Called by MAIN.

(ØUT2)
Entry point of subroutine ØUTl; prints the conditions at each point in the flow solutions. Called by FRØZEQ and PRTA.

PERT
Computes the perturbations in the flow variables during the nonequilibrium solution by the perturbation method. Called by NØNEQ.

PIØMEG
Computes the Maxwell-averaged electron-neutral momentum transfer cross section for given electron temperature $T_e$. Called by CØMM.

PRØP
Computes the conditions at a point in the frozen flow solution. Called by FRØZEQ.

PRTA
Organizes the output of conditions at each point in the nonequilibrium flow solution. Called by NØNEQ.

PUTQIN
Organizes the calculations of the cross sections used in computing the transport properties. Called by TRANSP.

QCØUL
Computes the factor 0.8 $Q_c$ in the Coulomb cross section for use in the transport property calculations. Called by PUTQIN.

QEX
Computes the cross sections for exchange interaction. Called by PUTQIN.

QEXP
Computes the cross sections for the exponential potential $\phi = Ae^{-E/\rho}$ . Called by PUTQIN.

QINTRP
Interpolates tabular data in cross section calculations. Called by QEXP, QLJ, and QTAB.

QLJ
Computes the cross sections for the Lennard-Jones (6-12) potential. Called by PUTQIN.
QMIX
Computes transport cross sections from the mixing rule, equation I(102).* Called by PUTQIN.

QREPP
Computes the cross sections for an attractive or repulsive power law potential $\Phi = Ar^{-\eta}$. Called by PUTQIN.

QSAME
Sets the cross sections for a species pair equal to a constant multiple of those computed previously for another pair. Called by PUTQIN.

QTAB
Computes the transport cross sections by linear interpolation in tables of $\Omega(1,1)$, $\Omega(2,2)$, and $B^*$ versus temperature. Called by PUTQIN.

Q11
Computes the transport cross sections using the fairing option, equations I(107) and I(108). Called by PUTQIN.

Q12
Computes the cross sections by the generalized mixing rule, equation I(109). Called by PUTQIN.

Q13
Sets one of the cross sections for a given species pair equal to a constant times another cross section for the same pair. Called by PUTQIN.

Q14
Recomputes the cross sections for a species pair using equations I(112). Called by PUTQIN and Q11.

RADIUS
Computes the profile ordinate, the geometric area ratio, and another quantity required in the calculation of the boundary layer at the first point in the flow solution. Called by BLAYER.

*Equations in Volume I (ref. 1) are cited in the form I(  ).
READ Reads the input data for each case and sets up arrays defining the flow geometry and gas model. Called by MAIN.

RESET In the IBM 360 version, simulates a UNIVAC 1108 library routine for storing the time at the start of a run. Called by ELTIME.

RESTMP Computes the equilibrium conditions in the upstream reservoir based on pressure and mass flow or mass flow and stagnation enthalpy data. Called by MAIN.

RNKT Computes the changes in the species concentrations and other flow variables over a step in the nonequilibrium integration. Called by NØNEQ.

SHOCK Computes the pressure ratio for an oblique shock in a perfect gas. Called by WEDGE.

SIMQ Solves systems of linear inhomogeneous equations. Called by DSMSØL and RESTMP.

STUNTS Prints an edit of species thermal properties. Called by MAIN.

(STUNT2) Entry point in subroutine STUNTS; prints edits of the steps in the transport cross section calculation. Called by MAIN.

THERM Computes the species thermal properties at a specified temperature. Called by DERIVS, EqCalc, PrøZeQ, Model, Newrap, and STUNTS.

(THERM1) Entry point in subroutine THERM; computes only the species specific heats at a specified temperature. Called by MAIN, Øutl, and TRANSP.

THRØAT Implements the switch from the upstream solution by the inverse method to the downstream direct integration. Called by NØNEQ.
(TIME) Entry point in subroutine RESET (IBM 360 version only); simulates a library timing routine available on the UNIVAC 1108. Called by ELTIME.

TRANSP Organizes the transport property calculations. Called by BLAYER, FRÖZ EQ, MAIN, MÖDEL, ØUT1, WEDGE, and STUNT2.

(TRANSX) Entry point in subroutine TRANSP; sets up arrays used in the transport property calculations. Called by MAIN.

WEDGE Computes and prints the conditions on the surface of a blunt wedge model inserted into the flow. Called by MÖDEL.

WESØIN Calculates the quantities $z$, $zz'$, and $(zz')'$, based on the approximate solution $I(48d)$ of the wedge equation, for given values of $\Gamma$ and $\Gamma'$. Called by WEDGE.

XSECT Sets arrays required for applying the precoded cross section data to the species present in the current gas model and sets up the edited cross section calculation. Called by TRANSX.

2.3 **UNIVAC 1108 Version**

The UNIVAC version of NATA has been set up to run on the 1108 processors at NASA/JSC under the EXEC II operating system. This version is a single precision program in which some arrays and three subroutines (SHØCK, SIMQ, and WESØIN) are typed as double precision to maintain adequate accuracy. These three routines all perform operations which, under some conditions, can lead to loss of many significant digits because of cancellation of terms. The typing of these routines as double precision is accomplished by retaining the IMPLICIT REAL*8 (A-h, Ø-Z) statements from the IBM 360 version. The UNIVAC Fortran V compiler interprets these statements correctly, a feature included by Sperry Rand in the interest of compatibility with IBM 360 Fortran IV.
NATA is too large to be run on the 1108 without use of overlay. Figure 1 shows the overlay structure employed. The overlay is implemented by means of the EXEC II Memory Allocation Processor (ref. 4). In figure 1, the names enclosed in boxes represent the routines of NATA listed in Section 2.2. Each box represents a segment or subsegment with the associated name (S.--.) given in parentheses. The level 0 segment S.MAIN is permanently in core. The various level 1 segments overlay one another. In a typical NATA case, the loading sequence for the level 1 segments is as follows:

S.READ  Read input data, set up gas model, etc.
S.LIST   Print problem summary.
S.INIT   Compute reservoir conditions, etc.
S.XSEC   Set up transport cross section calculation for current gas model.
S.OUT    Print definitions of output labels.
S.FRZ    Calculate equilibrium solution to throat, to obtain displacement thickness at throat.
S.INIT   Recalculate reservoir conditions corrected for displacement thickness at throat.
S.FRZ    Calculate frozen solution; calculate equilibrium solution.
S.NONE   Calculate nonequilibrium solution.

In the nonequilibrium solution, the subsegment S.N2 is loaded first for use in the initial solution by the perturbation method. When the nonequilibrium integration is started, the integration routine RNKT is loaded, overlaying S.N2.

Control cards for using NATA on the UNIVAC 1108 processors at NASA/JSC are shown in figures 2 and 3. Figure 2
Figure 1. Overlay Structure for UNIVAC 1108 Version of NATA
Z RUN NO3Z0...... SCOTT/  
N,MSG FILE REQ TAPE 01 FH432 0 FSTRN 00  
; C=OUT  
FOR MAIN, MAIN  
FOR SUB, SUB  
MAP NATA/NATA  
S.0 SEG S.MAIN-*(S.READ,S.INIT,S.XSEC,S.FROZ,S.NONE,S.OUT,S.LIST,  
S.STUN,S.DUMP)  
S.MAIN SEG MAIN-BLKD1-BLKD2-NRMAX-TRANSP-KINT-Q11-Q12-Q13-Q14-PUT0IN-  
QCQU-QEXP-QEX-QINTRP-QLJ-QNUM-QREPQ-QSAME-QTAB-FLCOND-KANDMU-  
OUTI-MODEL-PLAYER-WEDGE-GEOMAR-RADIUS-SHOCK-DUMP-TERM-WEGLN-  
NEXTMP-FINDX-A' OLNL-NEWRAP-DSMSOL-SIMQ-EGC\C-ELTIME  
S.READ SEG READ  
S.LIST SEG LIST  
S.STUN SEG STUNTS  
S.DUMP SEG DUMPEX  
S.INIT SEG INIT-RESTOP-INGAS-MATINV  
S.XSEC SEG XSECT-AXSECT-BXSECT-CXSFCT  
S.FROZ SEG FROZEQ-PROP  
S.NONE SEG NONEQ-PRTA-BLCALL-THROAT-EXACT-EPART-DERIVS-PIOWEG-GEOM-AESOLN-  
COMM-*(RNK1,S.N2)  
S.N2 SEG AXFIT-PERT  
S.OUT SEG OUT  
\XQT CUR  
\TRW C  
\JT C  
\CF C  
\REL C  

Figure 2. Control Cards for Generating a NATA PCF  
Tape on the UNIVAC 1108  

Z RUN NO3Z0...... SCOTT/  
N,MSG FILE REQ TAPE 01 FH432 0 FSTRN 00  
\ASG A="01234  
\XQT CUR  
\TRW A  
\IN A  
\REL A  
\XQT NATA  

Figure 3. Control Cards for Running NATA on the  
UNIVAC 1108  

ORIGINAL PAGE IS  
OF POOR QUALITY
illustrates a control card deck for generating a NATA PCF tape. The first card is a schematic run card. The deck is set up to compile MAIN and a subroutine "SUB"; the source decks are assumed to be inserted following the "FDR" cards. The "RUN", "MSG", "ASG", "FDR", "MAP", "XQT", "TRW", "OUT", "TEF", and "REL" cards all have a 7-8 punch in column 1 which was not reproduced by the equipment used to print this listing.

Figure 3 shows the control cards for a NATA run from an existing PCF. The "V01234" following the equals sign on the ASG card is a representative reel number for the PCF tape. The "XQT NATA" card should be followed by the input data for the case, which are described in Section 2 of Volume II (ref. 2).

The control cards required for use of NATA at other UNIVAC 1108 installations may differ in some respects from those illustrated in figures 2 and 3.

2.4 IBM 360 Version

Since NATA is run as a single precision program on the UNIVAC 1108, single precision function names (SQRT, ABS, AMAX1, etc.) are used throughout the program, except in the three double precision routines, SHOCK, SIMO, and WESOIN. To avoid the tedium of changing function names when the code is converted for use as a double precision program on the IBM 360, the single precision names are retained in the source program for the IBM version and are interpreted as referring to the double precision functions. In the IBM 360/75 installation at Avco Systems Division, this reinterpretation of the single precision Fortran function names is accomplished by using a special subroutine library in the link edit step. At other installations where such a special library may not be available, the same result can be achieved by including, in the NATA source deck, Fortran subroutines defining all of the single precision Fortran function names in terms of the corresponding double precision functions. An example of this procedure is provided by the function EXP in the IBM version of NATA. As shown by the listing of EXP in Section 5, this subroutine redefines EXP in terms of DEXP. It also
sets \( \exp(X) \) to 0 for \( X \) less than \(-180\), to avoid the production of paper-wasting underflow messages.

Since the Fortran G compiler gives a level 8 diagnostic when a single precision function name is used with a double precision argument, the G compiler cannot be used on the NATA version presented here. Instead, the H compiler, which gives only a level 4 diagnostic, must be used. Several of the NATA routines, including READ and the two block data routines, are too large for the standard H compiler. At Avco Systems Division, an available large compiler (FORTHMAX) is used to process the NATA source deck.

In both versions of NATA, the elapsed time since the beginning of the run is printed out at several stages in each case. In the UNIVAC version, these times are obtained by calling routines \texttt{RESET} and \texttt{TIME}, which are available in the 1108 system library. In the IBM 360 version in operation at Avco Systems Division, \texttt{RESET} and \texttt{TIME} are simulated using the system library routine \texttt{ACUCPU}, which gives the unexpended CPU time remaining before automatic job termination at \texttt{TIME.G0}.

In summary, conversion of the UNIVAC version of NATA into the IBM 360 version requires the following operations:

1. An \texttt{IMPLICIT REAL*8} (A-H, \( \phi-Z \)) card must be inserted into each NATA routine except \texttt{ELTIME}, \texttt{\phiUT}, \texttt{RESET}, \texttt{SHOCK}, \texttt{SIMQ}, and \texttt{WESLN}. In \texttt{MAIN}, this must be the first card apart from comment cards. In the other routines, the \texttt{IMPLICIT} statement must directly follow the \texttt{SUBROUTINE}, \texttt{FUNCTION}, or \texttt{BLOCK DATA} statement.

2. The subroutines \texttt{EXP} and \texttt{RESET} must be inserted.

3. If a special system library for interpreting the single precision Fortran function names as double precision functions is not available at the computer installation, Fortran subroutines to define the single precision names as double precision functions must be added to the source deck.
Figure 4 illustrates the control cards used to run NATA from a tape containing the binary deck, on the Avco Systems Division IBM 360/75 system. (The required JCL cards will differ somewhat at other IBM 360 installations). The fourth, fifth, and sixth cards in figure 4 invoke the special system library which interprets the single precision function names as referring to the corresponding double precision functions. The binary tape containing NATA is identified on the DD card with VOLUME = SER = NATA3. The data set GØ.FT08F001 is the binary tape upon which NATA can write data for subsequent plotting. GØ.FT07F001 is the data set for punched output.
// JOB
// MSGLEVEL=1  
// XEC FORTHLG, REGION.GO=420K, TIME.GO=4  
// LKED, SYSLIB DD DSNAME=VER1, TEMPLIB, DISP=SHR  
// DD DSNAME=S'S1, FORTLIB, DISP=SHR  
// DD DSNAME=S'S1, DOUBLEP, DISP=SHR  
// LKED, SYSUT1 DD SPACE=(1024, (600, 50))  
// LKED, SYSLMOD DD SPACE=(3072, (200, 10, 1), RLSE)  
// LKED, SYSLIN DD UNIT=(TAPE9, , DEFER), LABEL=(*BLP), DISP=(OLD, KEEP),  
// DCB=(RECFM=FB, LRECL=80, BLKSIZE=3200), VOLUME=SER=NATA3  
// DD *  
// ENTRY MAIN  
//  
// GO.FT08F001 DD UNIT=(TAPE9, , DEFER), LABEL=(*BLP), DISP=(NEW, KEEP),  
// DCB=(RECFM=VB5, LRECL=92, BLKSIZE=4604), VOLUME=SER=DATASV  
// GO.FT07F001 DD SYSOUT=B  
// GO.FT05F001 DD *  

Figure 4. Control Cards for Running NATA on the IBM 360
3. ANALYSIS OF ROUTINES

The present section discusses and explains the main program and each subroutine of the NATA code. The purpose of this exposition is to provide an entry into the coding for programmers who wish to analyze errors, make corrections, or introduce modifications into NATA.

All of the physics and gas dynamics and most of the mathematical analysis underlying the programming have already been documented in Volumes I and II of this report (refs. 1, 2). Where appropriate, individual sections and equations of Volumes I and II are referenced.* In cases where the underlying mathematical analysis has not been documented previously, it is explained in the present section.

The routines of NATA are discussed in the following order: first the main program, then all subroutines and functions in alphanumeric order of their names. This is the same order as that in which the routine names are listed in Section 2.2 and at the beginning of Section 5, and in which the source listings are presented in Section 6.

3.1 Main Program

The main program performs certain initialization functions, calls the major subroutines in proper sequence, implements some of the input options, and controls job termination. A flowchart of MAIN is shown in figure 5. The operations shown in this flowchart will be discussed approximately in the order in which they are performed during execution of the code.

3.1.1 Job Initialization

The 12 input control parameters ISWIA to ISW6B are initialized by arithmetic statements in MAIN before any input data

*Equations from Volumes I and II will be referenced in the form I( ) and II( ).
Figure 5a. Flowchart of NAPA Main Program (Part a)
have been read. This procedure is used because these parameters are in unlabelled common, and hence cannot be preset in a BLOCK DATA routine.

3.1.2 Case Initialization

The calculations for each case in a job always begin with statement 10. First, subroutine ELTIME is called to determine the elapsed time ET0 since the start of the first case. Then subroutine READ is called. This routine reads the input data for the case, sets up arrays describing the gas model and the nozzle geometry, and prints some of the data defining the case being run. Next, subroutine LIST is called to print the remaining data defining the current case. Subroutine ELTIME is then called again to print the time. ELTIME is called at various locations in MAIN, to provide data on the execution time consumed by major portions of the calculation. These subsequent calls to ELTIME are omitted from the flowchart in the interest of simplification.

Before the call to LIST, the value of the logical input variable N\text{T\text{RAN}} is checked. N\text{T\text{RAN}} is a switch to allow the suppression of all transport property calculations throughout NATA. With N\text{T\text{RAN}} = .\text{\text{\text{\text{TRUE}}}}., flow solutions can be run for a gas model containing user-specified species for which transport cross section data have not been provided. Of course, when the transport property calculations are suppressed, the boundary layer cannot be included in the flow solutions. Hence, for N\text{T\text{RAN}} = .\text{\text{\text{\text{TRUE}}}}., ISW3B is set to zero. However, the input value of ISW3B is saved in the location IS3, and is reset before the call to READ in subsequent cases.

MAIN next increments the case counter NCASE and initializes some variables including the indicator ERR in common block /\text{ERRPXR}/. This indicator is set to .\text{\text{\text{\text{FALSE}}}}. near the beginning of each case. If the programmed checks in any subroutine subsequently detect an error, ERR is set to .\text{\text{\text{\text{TRUE}}}}., and this indication causes control to be returned to MAIN as quickly as possible, with no further calculations. Statements in MAIN then cause subroutine DUMPEX to be called. DUMPEX prints out a diagnostic dump consisting of the current values of most of the variables in labelled and unlabelled
common. The details of this scheme are explained more fully in the sections (below) on subroutines DUMP and DUMPEX. Most of the tests on ERR in MAIN have been omitted from the flowchart for brevity, but the ones following CALL FRÆZEN, CALL EQUIL(2), and CALL NØNEQ have been included to illustrate the method.

MAIN next calls subroutine INGAS to set up certain arrays used to describe the chemical nature of the gas mixture. Then, if ISW2B ≤ 0 (i.e., if the determination of reservoir conditions is to be based partly on the total mass flow), subroutine RESTMP is called to compute the reservoir temperature and pressure. Unless ISW2B is negative, subroutine INIT and INTA are then called. INIT initializes a number of control variables and nondimensionalizes some species properties using the reservoir temperature. INTA computes the species mole fractions and the thermal properties of the gas in the reservoir. For negative ISW2B, INIT and INTA are called in RESTMP with the final reservoir temperature and pressure, and thus do not have to be called in MAIN.

3.1.3 Special Options

If ISW6A is negative, NATA computes and prints tables of thermal properties for all of the species in the current gas model, instead of calculating a flow solution. This operation is performed by subroutine STUNTS.

If ISW6A is positive, NATA computes and prints out the reservoir conditions (as usual), but does not proceed with the flow solutions. For the specific value ISW6A = 2, the calculation of the transport properties in the reservoir is also omitted. This feature allows the NATA user to employ the code for thermochemical calculations for a gas model containing nonstandard species without setting up transport cross section data for the species.

If ISW6A is 0 (the normal, preset value), NATA generates one or more flow solutions.
If ISG6A is zero or positive (but not equal to 2), the transport properties in the reservoir are computed. Before transport property calculations can be done, the entry TRANSX of subroutine TRANSP must be called. However, the call to TRANSX can be omitted if TRANSX has been called in a previous case with the same gas model. The gas model is the same as in the preceding case if SUPG@ = .TRUE., and TRANSX was called in a previous case if IRC@ = 0. If N@TRAN is .TRUE., the call to TRANSX is skipped.

Another special option is an edit of the transport cross section calculations, performed by calling the entry STUNT2 of subroutine STUNTS. This call is executed if the input control parameter ISW1B is nonzero. For ISW1B negative, STUNT2 also prints a table of averaged transport cross sections; in this case, no flow solution is run.

### 3.1.4 Specific Heat of the Gas at the Wall Temperature

MAIN computes the specific heat of the gas at the nozzle wall temperature, TWALL. This calculation is based on the assumption that the gas at this low temperature is a mixture of the "cold species" used to define the elemental composition. The specific heat is calculated as

\[ C_{pw} = \frac{R_0}{W_C} \sum_{k=1}^{n_C} X_k \frac{C_{pk}}{R_0} \]  

(1)

in which \( W_C \) is the mean molecular weight of the cold gas, \( n_C \) the number of cold species, \( X_k \) the mole fraction of the \( k \)th cold species, and \( C_{pk}/R_0 \) the nondimensional molar specific heat at constant pressure for the \( k \)th cold species. This species specific heat is given by CCPJ(I), where \( I = IJCS(K) \) is the index, in the list of species for the current gas model, of the \( k \)th cold species.

### 3.1.5 Definitions of Output Variable Labels

In the first case of each job (NCASE = 1), MAIN calls subroutine @OUT to print a list of definitions, with units,
of the output variable labels used in printing out the flow conditions and test conditions on models. These labels differ, in most cases, from the Fortran variable names used in the code.

3.1.6 Correction of the Reservoir Conditions for the Displacement Thickness at the Throat

If the boundary layer is to be included in the solution (ISW3B ≠ 0), and if the reservoir condition calculations are based partly on the total mass flow (ISW2B < 0), then NATA normally recalculates the reservoir conditions, taking account of the effects of the boundary layer displacement thickness on the equilibrium sonic mass flux. For ISW2B ≤ 0, subroutine RESTMP determines the reservoir temperature and pressure so as to match the sonic mass flux SMASS and one other condition (on pressure or enthalpy). Initially, when RESTMP is called the first time, SMASS is calculated as the ratio of the input total mass flow to the geometric cross sectional area of the nozzle at the throat. To correct these results for the displacement thickness effect, MAIN calls entry EQUIL(IPASS) of subroutine FRZEQ with the argument IPASS = 1. EQUIL then computes the equilibrium solution up to the throat, including the boundary layer. MAIN then calculates the effective cross sectional area ATH at the throat including the effects of the displacement thickness, recomputes the sonic flux SMASS, and calls RESTMP for a second time to recompute the reservoir conditions. This correction can be suppressed, if it is not desired, by setting MFITER = 0 in the input.

3.1.7 Conversion Factors

Next, MAIN sets the array of conversion factors CF(I) in common block /C0NVRT/. These factors convert the non-dimensional flow variables used in the internal computations to the units used in the printed output.

3.1.8 Output of Reservoir Conditions

The nondimensional flow variables CT = T/T₀, PRES = p/p₀, etc., are now set to their reservoir values and subroutine OUT1 is called to print out the reservoir conditions.
3.1.9 Solution Header Records on Tape 8

If the frozen flow solution is to be generated (ISW1A ≠ 0) and if data are to be written on a binary output tape for subsequent plotting (DATAPE = .TRUE.), the main program next writes a "header" record on tape 8 for the frozen solution. This record contains the nozzle index NOZZLE = NPRFL(1), the type of solution (ISOLN = 1 for frozen flow), and the reservoir conditions.

3.1.10 Frozen Solution

To generate the frozen flow solution, MAIN calls the entry FRZFN of subroutine FRZEQ. It then calls OUTPUT to print the sonic-point conditions in the frozen solution.

3.1.11 Equilibrium Solution

MAIN calls subroutine NRMAX to compute the conditions at the sonic point in the equilibrium solution. These data are needed for both the equilibrium and nonequilibrium solutions. Then, unless ISW3A = 0, MAIN checks DATAPE. If this control variable is .TRUE., a header record is written on tape 8 for the equilibrium solution, and the entry EQUIL(IPASS) of subroutine FRZEQ is called with IPASS = 2 to produce the solution. Finally, OUTPUT is called to print out the equilibrium throat conditions.

3.1.12 Nonequilibrium Solution

If the nonequilibrium solution is to be computed (ISW2A ≠ 0), MAIN determines the constants C and α in the area-density relation I(383) assumed in the solution by the inverse method upstream of the throat. The parameter α is determined by a Newton-Raphson solution of equation I(385) or I(386). The iteration is started using the initial estimate

\[ \alpha \approx 4 + 8 \ln \left( \frac{\rho}{\rho_0} \right) \]  \hspace{1cm} (2)

which can be derived by taking the logarithm of I(385) and
expanding $\ln (\alpha + 2)$ around $\alpha = 0$:

$$\ln (\alpha + 2) \approx 2 + \frac{\alpha}{2} - \frac{\alpha^2}{8} \quad (3)$$

Then to obtain starting conditions for the perturbation solution, MAIN sets the nondimensional temperature CT to 1-DELT, calls subroutine NEWRAP to compute the equilibrium flow conditions at this temperature, computes the corresponding effective area ratio AFNTS, and calls subroutine FINDX to determine the axial coordinate CX at which these conditions occur. If DATAPE = .TRUE., a header record is written on tape 8 for the nonequilibrium solution. Then subroutine NONEQ is called to generate the nonequilibrium flow solution.

3.1.13 Diagnostic Dumps

If a .TRUE. value of the error indicator ERR is detected anywhere in MAIN, control is immediately transferred to the statement CALL DUMPEX, which prints out the current values of a large number of common variables as an aid to identifying the cause of failure. If data have been written on tape 8, the tape is backspaced to eliminate the records written during the failed case.

3.1.14 Job Termination

At the end of the calculations for each case, a line of output is written containing the case number, the number of records written on tape 8 (if any), and the total number of records written thus far in the job. Then the value of the job termination control parameter ISW4A is checked. If ISW4A is nonzero, the input data for the next case are read. If ISW4A is zero, a summary of the completed and failed cases in the job is written and, if DATAPE is .TRUE., an end file mark is written on tape 8 and the tape is rewound.

3.2 Subroutine AESLN

This subroutine calculates the effective area ratio at a given position, X, in the nozzle. The value X is a subroutine
argument. Other input data to the routine are contained in the common block `/AEGE@/`, namely, the geometric area ratio $S_1$, the derivative of the geometric area ratio $S_2$, and the square root of the geometric area ratio $\text{SQRTA}$. To set $S_1$ and $S_2$, a call to AES@LN(X) must be preceded by a call to GE@MAR(X,$S_1$,$S_2$). In addition, if NPRFLS = 1 (the flow is in a nozzle, not a channel) and JDIM = 1 (the nozzle is axisymmetric), then $\text{SQRTA}$ must be computed in the calling routine before AES@LN is called.

The calculations of the effective area ratio are based upon equations I(126), I(130) and I(134), respectively, for the cases of a two-dimensional nozzle (NPRFLS = 1, JDIM = 0), an axisymmetric nozzle (NPRFLS = 1, JDIM = 1), and a rectangular channel (NPRFLS = 2). The effective area ratio obtained is stored in the location $S_1$ of common block `/AEGE@/`, replacing the geometric area ratio.

In addition, the subroutine computes the derivative, $dA_e/dx$, of the effective area ratio, using the following formulas:

**2D Nozzle**

\[
\frac{dA_e}{dx} = \frac{\frac{dA_g}{dx} - \frac{1}{Y_0} \frac{dS^*}{dx}}{1 - \frac{S^*}{Y_0}}
\]  
\( (4) \)

**Axisymmetric Nozzle**

\[
\frac{dA_e}{dx} = \sqrt{A_e} \left[ \frac{\frac{dA_g}{dx} - \frac{2}{Y_0} \frac{dS^*}{dx}}{1 - \frac{S^*}{Y_0}} \right]
\]  
\( (5) \)

**Channel**

\[
\frac{dA_e}{dx} = \frac{(z - S^*_2)(\frac{dy}{dx} - \frac{dS^*_2}{dx}) + (y - S^*_1)(\frac{dz}{dx} - \frac{dS^*_2}{dx})}{(y_0 - S^*_1)(z_0 - S^*_2)}
\]  
\( (6) \)
These relations can easily be derived from equations I(126), I(130), and I(134). The computed value of $dA_e/dx$ is stored in the location S2 of common $/AEGE0M/$, where it replaces the derivative of the geometric area ratio.

3.3 Subroutine AGS9IN

The purpose of this subroutine is to solve for the geometric area ratio $A_g$ when the effective area ratio $A_e$ and boundary layer displacement thickness $\delta^*_i$ are given. These input data are provided in the argument list: $A_e = A_E$, $\delta^*_i = \text{DEL}(i)$, where the index $i$ runs from 1 to 2 in the case of a channel. The argument list also includes the indicator $\text{UPDWN}$, which specifies whether an upstream (-1.) or downstream (+1.) solution is desired. The computed value of $A_g$, and the corresponding value of $x$, are returned to the calling routine through the arguments $AG$ and $X$, respectively.

The solution for $A_g$ is straightforward and explicit in the case of a nozzle. It is given by equation I(127) if the flow geometry is two-dimensional and by I(131) if it is axisymmetric. In both cases, the value of $X$ corresponding to $A_g$ is determined by calling FINDX.

In the case of a channel, the relation between $A_e$ and $A_g$ is given by equation I(135) and the profile functions $y(x)$, $z(x)$. This relation does not permit an explicit solution for $A_g$. In AGS9IN, a numerical Newton-Raphson method is used to determine $A_g$. Equation I(135) is written as

$$A_g = A_c + \frac{y \delta^*_2 + z \delta^*_1}{y_0 z_0}$$

where

$$A_c = (1 - \frac{\delta^*_1}{y_0}) (1 - \frac{\delta^*_2}{z_0}) A_e - \frac{\delta^*_1}{y_0} \frac{\delta^*_2}{z_0}$$

$y_0$, $z_0$
The quantity \( A_c \) and the factors \( S_1/y_0z_0 \) and \( S_2/y_0z_0 \) are constants during the solution for a particular call to the subroutine. From (7), the object of the calculation is to find a value of \( x \) such that \( F(x) = 0 \), where

\[
F(x) = A_c + \frac{y(x) S_2^* + z(x) S_1^*}{y_0z_0} - A_g(x) \quad (9)
\]

Let \( x, F \) denote the values of these quantities from the current iteration and \( x_0, F_0 \) the values for the previous iteration. Then, based on a linear approximation to (9), the value of \( x \) required to make \( F(x) = 0 \) is estimated to be

\[
x_n = x - F \cdot \frac{x - x_0}{F - F_0} \quad (10)
\]

The iteration based on this relation begins at statement 60 in the subroutine. To start the iteration, \( A_c \) is set equal to \( A_{c0} \), and the corresponding \( x \) is computed using FINDX. The second point is obtained by computing \( A_g \) from (7) and again using FINDX. Equation (10) is then used in the third and all subsequent steps. The convergence criterion is that \( |F|/A_g \) be less than or equal to \( 10^{-5} \), or that \( F = F_0 \).

If \( A_g(x) \) in (9) is expressed by means of equation I(120), equation (9) can be rewritten in the form

\[
F(x) = (1 - \frac{S_1^*}{y_0})(1 - \frac{S_2^*}{z_0}) A_e - \frac{y}{y_0} \frac{x - S_1^*}{z_0} - \frac{z - S_2^*}{z_0} \quad (11)
\]

If \( y \) and \( z \) both have minima at the throat \( x = 0 \), as is normally the case in channels, then the maximum value of \( F(x) \) occurs at \( x = 0 \) and is equal to

\[
F_{\text{max}} = F(0) = (1 - \frac{S_1^*}{y_0})(1 - \frac{S_2^*}{z_0}) A_e - (1 - \frac{S_1^*}{y_0})(1 - \frac{S_2^*}{z_0}) \quad (12)
\]
In order for a solution of the equation \( F(x) = 0 \) to exist, it is necessary that \( F_{\text{max}} \) be greater than or equal to zero. This is always true if \( \delta_1^* = \delta_1^\# \) and \( \delta_2^* = \delta_2^\# \). Also, in practice, it is always true when \( A_e \gg 1 \). However, when \( A_e \) is nearly equal to 1 and \( \delta_1^* < \delta_1^\# \), \( \delta_2^* < \delta_2^\# \), \( F_{\text{max}} \) can be negative. These circumstances can arise near the sonic point (where \( A_e = 1 \)) when the sonic point is displaced from the geometric throat by the presence of large gradients in the displacement thicknesses. When this occurs, the iteration formula (10) extrapolates past the throat and gives an \( x_n \) of the wrong sign, i.e., the opposite sign to \textsc{updown}.

When such a condition is detected, \textsc{agsoln} attempts to "fix" the problem by resetting the displacement thicknesses \( \delta_1^* \), \( \delta_2^* \) to their values at the sonic point, \( \delta_1^\# \) and \( \delta_2^\# \), respectively, and restarting the iterative solution at the beginning. Under the circumstances in question, this adjustment can have little effect on the value of \( A_y \) computed (which is near unity). Equation (12) shows that after this replacement, \( F_{\text{max}} \) is the product of a positive factor and \( (A_e - 1) \), and hence is \( \geq 0 \). Thus, if the problem was indeed caused by the combination of circumstances described above, the "fix" should guarantee the existence of a solution of the equation \( F(x) = 0 \). The coding provides a dump and an error exit should the problem recur after the "fix" has been carried out. In this case, the problem has a different cause, such as an error in a user-supplied channel profile curvefit.

3.4 Subroutine \textsc{axfit}

This subroutine solves for the value of \( x \) corresponding to a given effective area ratio (\textsc{afnts}) during the initial (perturbation) portion of the nonequilibrium solution. If the solution is being calculated without the boundary layer (\textsc{isw3b} = 0), then the geometric area ratio is equal to the effective area ratio, and \( x \) (denoted by \textsc{cx}) is determined by calling subroutine \textsc{findx}. If the boundary layer is included in the solution, but subroutine \textsc{thrgat} has not yet been called to reset \textsc{iupd} from 1 to 0, the effects of the boundary layer displacement thickness on the effective flow geometry are neglected, and \( x \) is again obtained from \textsc{findx}. However, if the boundary layer is included and \textsc{iupd} = 0, then the inviscid flow is coupled to the boundary layer and...
x is accordingly determined by calling AGS@LN. Both FIND\^x and AGS@LN require an indicator which specifies whether the upstream (UPD\^\@WN = -1) or downstream (UPD\^\@WN = 1) solution for the x corresponding to the effective area ratio AFNTS is desired. In AXFIT, this indicator is determined by testing whether the nondimensional temperature CT is greater than or less than its value CMAX at the geometric throat based on the equilibrium flow solution.

3.5 Subroutine AXSECT

Subroutine AXSECT is called by subroutine XSECT for the purpose of adding data for additional species pairs to the edited cross section data in common block/TRANS7/. The first argument LL of the subroutine gives the step in the cross section computations at which the additional pairs are to be added, while the second argument NN gives the total number of species to be considered in adding the data. The indices of the actual species to be considered in the added data are then given, in numerical order, by the first NN entries of the variable I in common block/TRANS2/. Subroutine AXSECT searches through all pairs (II, JJ) of these species with II ≤ JJ, to find those for which cross section data have not been previously specified, as indicated by the fact that Q(I,II, JJ) = 0. All such pairs are then added to the IQ, JQ arrays, starting at the location NQ(LL) + 1, with the original data in these locations being moved back to make room for the new values. Further, the subroutine sets Q(I,II, JJ) = 1. for all pairs added to the IQ, JQ arrays and revises the NQ array to be consistent with the new IQ, JQ values, assuming that all of the added pairs are to be included in the LL\textsuperscript{th} step of the cross section computations.

3.6 Subroutine BLAYER

The boundary layer calculations in NATA are carried out by the method described in Section 5 of Volume I (ref. 1). These calculations are performed by subroutine BLAYER. The boundary layer properties are calculated from the integral I, equation I(172). This integral is used to determine the correlation parameter n, equation I(174). The momentum thickness
\( \Theta \), displacement thickness \( \delta^* \), shear stress \( \tau_w \), and heat flux \( q_w \) are then calculated from \( n \) and other quantities.

The overall structure of BLAYER is diagrammed in figure 6. The calculations are all contained in a \( \text{D} \) loop with index \( L \) running from 1 to NPRFLS. In the case of a nozzle (NPRFLS=1), the calculations are executed only once. For a channel (NPRFLS=2), they are executed twice to compute the boundary layers on the two pairs of channel walls.

The subroutine argument FINAL is a logical variable specifying whether the new flow point is an actual point of the solution or merely an intermediate step in the Runge-Kutta integration used to generate the nonequilibrium flow solution. For FINAL = .FALSE., the important variables used in BLAYER are restored to their values at the beginning of the step before the RETURN is executed, and the calculations of shear stress \( \tau_{wA} \) and heat flux \( Q_{WDT} \) are omitted.

\( \text{IP} \) is a counter for the flow points at which boundary layer calculations are done in each flow solution. \( \text{IP} \) is initialized to 1 in subroutine \( \text{FROZEQ} \) for frozen and equilibrium solutions and in subroutine \( \text{NONEQ} \) for nonequilibrium solutions. Each time BLAYER is called with FINAL = .TRUE., \( \text{IP} \) is incremented by 1 before the RETURN.

Some of the physical variables which are used frequently are given short mnemonic names which are equivalenced to their standard names in unlabelled common

3.6.1 Initializations

When BLAYER is called for the first time in a given flow solution (\( \text{IP} = 1 \)), a number of quantities are set or initialized by the statements down to 90. These quantities include the coefficient \( B \) in the linear relation \( I(164) \) between the momentum parameter \( N \) and the correlation parameter \( n \), and the transport properties \( \text{PRW} = Pr_w \) and \( \text{VISCW} = \mu_w \) at the wall temperature. The wall is assumed to be catalytic; hence, these transport property calculations are based on the mole fractions \( Q_PJ(K) \) for the cold gas mixture. This section of the routine also sets the indicator
Figure 6. Flow Chart of Subroutine BLAYER
ITYPE, which specifies the type of nozzle geometry:

- ITYPE = 1  Two-dimensional nozzle
- ITYPE = 2  Axisymmetric nozzle
- ITYPE = 3  Rectangular channel

3.6.2 Transport Property and Geometry Calculations

The section of BLAYER from statement 90 down to 270 computes the values of various physical quantities at the current flow point, including the transport properties and geometric parameters entering into the boundary layer calculation.

BLAYER uses the viscosity and Prandtl number at three temperatures, the wall temperature TWALL, the free-stream temperature, TE, and the reference temperature TREF. The calculations for TWALL are done only once per solution because the wall temperature is assumed constant. To save computing time, the transport property calculations at TE and TREF are not repeated at every flow point, but are done only when the temperature or the gas mole fractions have changed significantly. The criterion for performance of the transport property calculations is a change of 0.01 in any of the mole fractions or a change of 0.5 percent in the temperature. The mole fractions at which calculations were last done are represented by XTEST(J). The corresponding temperatures are TEP and TREFP.

The factor \( r_{2j} \) in equation \( I(171) \) is denoted by AJ in BLAYER. Because of the form of equations \( I(171) \) to \( I(173) \), \( r_{2j} \) can be replaced by a constant times \( r_{2j} \) without affecting the computed results for the boundary layer thicknesses, the heat flux, and the shear stress. For two-dimensional nozzles, AJ is set to 1, which is the value of \( r_{2j} \) for \( j = 0 \). For axisymmetric nozzles, AJ is set to the geometric area ratio,

\[
A_g = \left(\frac{y}{y_0}\right)^2
\]

(13)
which is $r^2/\gamma_0$. In the case of a rectangular channel, for each profile, $A_J$ is set to the square of the radius of the other profile, since this radius is proportional to the width of the surface whose boundary layer is being calculated.

For nozzles, the cosine of the angle $b$ between the profile tangent and the axis is calculated from $dA_g/dx$. In general,

$$\cos b = \frac{1}{\sqrt{1 + \tan^2 b}} = \frac{1}{\sqrt{1 + (dy/dx)^2}}$$

(14)

where $y$ is the profile ordinate. In the case of a two-dimensional nozzle, from equation I(118),

$$\frac{dy}{dx} = y_0 \frac{dA_g}{dx}$$

(15)

and hence

$$\cos b = \frac{1}{\sqrt{1 + y_0^2 (dA_g/dx)^2}}$$

(16)

In the axisymmetric case, from I(116),

$$\frac{dy}{dx} = \frac{y_0}{2\sqrt{A_g}} \frac{dA_g}{dx}$$

(17)

and thus

$$\cos b = \frac{1}{\sqrt{1 + \frac{y_0^2}{4A_g} \left( \frac{dA_g}{dx} \right)^2}}$$

(18)

The derivative $dA_g/dx$ in (16) and (18) is obtained by calling subroutine GE\O MAR. In the case of a channel, $dy/dx$ is computed by calling the entry GMAR3 of GE\O MAR, and $\cos b$ is evaluated directly from (14).
The viscosity-temperature exponent $\omega$, defined by equation I(191), is determined by fitting I(191) to the viscosity at the wall and reference temperatures. The coefficient $A$ in I(164) (denoted by $E_1$) is calculated from I(201) at the first flow point ($IP\Omega INT = 1$), and then held constant through the solution.

The integrand $F$ of I(172) is represented by $\phi$RDIN(L). Actually, $\phi$RDIN(L) differs from equation I(171) by a constant factor, which cancels out in the calculation of $n^I$ I(173) and thus has no effect on the boundary layer properties.

3.6.3 Integration from $x_0$ to First Flow Point

In general, the flow point at which $BLAYER$ is first called lies upstream of the throat. The Mach number at this point is low. However, the point may be well downstream of the point $x_0$ at which boundary layer development is assumed to start (see Sec. 5.10 of Volume I). Accordingly, the interval from $x_0$ to the first point of the flow solution is not treated as a single step for the purpose of the boundary layer calculation. Instead, when $BLAYER$ is called with $IP\Omega INT = 1$, the integral I(172) (denoted by $BLINT(L)$) and the streamwise boundary layer coordinate $\xi$ (denoted by $XI(L)$) are evaluated by a 10-step numerical integration from $x_0$ to the first flow point.

In the region covered by this integration, the Mach number is low so that the flow can be approximated as incompressible. Also, the boundary layer is thin in this region, in comparison with the nozzle or channel radius, so that the effective area ratio can be approximated by the geometric area ratio. Thus, the continuity equation $\rho u A_e' = \text{constant}$, can be approximated

$$M A_g = \text{constant} \quad (19)$$

The constant in (19) is represented by $CC$. At each point in the integration from $x_0$ to $x_1$, the geometric area ratio $A_g = AG1$ and the quantity $AGj$ corresponding to $r^2j$ are calculated by calling subroutine $RADIUS$, $M$ is obtained from (19), and a quantity $\phi$RD1 proportional to $\phi$RDIN(L) is computed with the
approximation $a_0/a_0 \approx P_e/P_0 \approx 1$. The increment in $\xi$ is computed using the Pythagorean theorem:

$$
(\Delta \xi)^2 = (\Delta x)^2 + (\Delta y)^2
$$

where $\Delta y$ is the change in the profile ordinate due to a change $\Delta x$ in the axial coordinate. BLINT(L) is calculated using trapezoidal-rule integration. At the first flow point, $\text{DLM}(L) = d/\sqrt{M}/dx$ is evaluated from (19).

3.6.4 Derivatives of the Mach Number

The derivative $dM_e/dx$ of the flow Mach number is an important variable in the boundary layer calculation because it represents the effect of a streamwise pressure gradient. The related quantity $dM_e/d\xi$ appears in the formula I(174) for the correlation parameter, $n$. All of the boundary layer properties depend upon $n$.

The derivative $dM_e/d\xi$ differs between the two profiles of a channel because the boundary layer coordinate lies along the channel surface and the two profiles have different inclinations to the nozzle axis. The value of $d/ln M_e/d\xi$ for the $L^{th}$ profile is denoted by $\text{DLM}(L)$. It is determined by first computing $d/ln M_e/dx$ and then correcting for the inclination of the nozzle or channel surface to the axis. For each flow point beyond the first, $d/ln M_e/dx$ is calculated from rate data if the flow solution is of nonequilibrium type and the numerical integration technique is being used (INEQ = 1). Otherwise, if the flow solution is of frozen or equilibrium type or if the perturbation technique is being used in a nonequilibrium solution (INEQ = 0), $d/ln M_e/dx$ is calculated by numerical differentiation. If the inviscid solution is being calculated by nonequilibrium integration, $\text{DLM}(L)$ is calculated from equations I(420), I(421), and I(291) or I(422). In the case of a chemical nonequilibrium gas model (NT = 1), the stagnation enthalpy $h_0$ is constant and hence I(421) and I(291) give

$$
\frac{du}{dx} = -\frac{1}{u} \sum_{j=1}^{n} \left[ \gamma_j \frac{dH_j}{dx} + H_j \frac{d\gamma_j}{dx} \right]
$$
The variables used in the code are

\[ \text{SHJ}(j) = \frac{H_i}{R_i T_0} \]  
(22a)

\[ \text{CCPJ}(j) \cdot DT = \frac{1}{R_0 T_0} C_p \text{pj} \frac{dT}{dx} = \frac{1}{R_0 T_0} \frac{dH_j}{dx} \]  
(22b)

\[ SU = u \sqrt{\frac{W_0}{R_0 T_0}} \]  
(22c)

\[ GJ(j) = \gamma_j \]  
(22d)

Hence (21) is coded as

\[ \frac{d(SU)}{dx} = - \frac{W_0}{SU} \sum_{j=1}^{n} \left[ GJ(j) \cdot \text{CCPJ}(j) \cdot DT + \text{SHJ}(j) \cdot \text{DGJ}(j) \right] \]

\[ = - \frac{\text{CMA} \cdot (\text{SCPG} \cdot DT + \text{SHJDGJ})}{SU} \]  
(23)

The sums denoted by SCPG and SHJDGJ are obtained from common. Similarly, in the case of an electronic nonequilibrium model \((NT = 2)\), equations I(421) and I(422) give

\[ \frac{dDX}{SU} = \frac{\text{DCHA}}{SU} - \frac{\text{CMA}}{SU} \left[ GJ(1) \cdot \text{CCPJ}(1) \cdot DT \text{E} \right. \]

\[ + \text{SCPGH} \cdot DT + \text{SHJDGJ} \]  
(24)

where

\[ \text{DCHA} = \frac{W_0}{R_c T_0} \frac{dh_0}{dx} \]  
(25)

\[ \text{SCPGH} = \sum_{j=2}^{n} GJ(j) \cdot \text{CCPJ}(j) \]  
(26)
In (24), quantities subscripted (1) refer to the electron species.

For equilibrium and frozen solutions, and for nonequilibrium solutions which are being developed by the perturbed equilibrium flow technique (INEQ = 0), the quantities SCPG and SHJDGJ are not available. In these cases, DLM(L) is calculated by numerical differentiation. For IPGINT = 2, a simple first-order difference expression

\[
\frac{dM}{dx} = \frac{M - M_p}{\Delta x}
\]

is used, where M is the Mach number at the current point and \( M_p \) that at the previous point. For IPGINT \( \geq 3 \), the following second-order difference calculation is used: Let \((M,x)\) be the data for the current point \((M_p,x_p)\) those for the previous point, and \((M_{pp},x_{pp})\) those for the point preceding the previous one. A quadratic fit to these data can be written in the form

\[
M = c_0 + c_1 x + c_2 x^2
\]

from which

\[
\frac{dM}{dx} = c_1 + 2c_2 x
\]

Substitution of the three data points into (28) gives, after elimination of \( c_0 \),

\[
\frac{M - M_p}{x - x_p} = c_1 + c_2 (x + x_p)
\]

(30a)

\[
\frac{M - M_{pp}}{x - x_{pp}} = c_1 + c_2 (x + x_{pp})
\]

(30b)
Hence
\[ c_2 = \frac{M-M_p}{x-x_p} - \frac{M-M_{pp}}{x-x_{pp}} \times \frac{x_p - x_{pp}}{x_p - x_{pp}} \]  
(31)

Now that \( c_2 \) is known, \( c_1 \) can be obtained from (30b);
\[ c_1 = \frac{M-M_p}{x-x_p} - c_2 \times (x + x_p) \]  
(32)

\( dM/dx \) is then calculated from (29), and \( DLM(L) \) is obtained from
\[ \frac{d\ln M}{dx} = \frac{1}{M} \frac{dM}{dx} \]  
(33)

3.6.5 Calculation of Boundary Layer Thicknesses

Next, the increase in \( \frac{d}{d} \) in the step from the previous flow point to the current one is calculated,
\[ \Delta \frac{d}{d} = \frac{\Delta x}{\cos b} \]  
(34)

and \( \frac{d}{d} \) is incremented by \( \Delta \frac{d}{d} \). The boundary layer integral \( I \) (equation I(172)), denoted by \( BLINT(L) \), is updated using the trapezoidal integration rule. The derivative \( d\ln M/dx \) is converted into \( d\ln M/d\frac{d}{d} \) by
\[ \frac{d\ln M}{d\frac{d}{d}} = \cos b \frac{d\ln M}{dx} \]  
(35)

Then \( n' \), equation I(173), is evaluated, and the correlation parameter \( n \) is calculated from I(174). The characteristic length \( L \) in these equations is taken to be \( R_0 \), defined as the profile radius in the case of a nozzle, and defined by
\[ \pi (R_0)^2 = \text{throat area} \]  
(36)
in the case of a channel. The momentum thickness is calculated from \( I(184) \). If the flow geometry is that of an axisymmetric nozzle, the transverse curvature correction \( I(187) \) is applied. The smoothed correlation parameter \( \bar{n} \), denoted by \( XSh(L) \), is computed from equation \( I(217) \).

The effect of free stream Mach number upon the correlations of boundary layer properties is contained in the dependence upon the hypersonic parameter \( \sigma \), equation \( I(190) \). The correlations used in BLAYER are based upon the Dewey-Gross calculations (ref. 5) for a perfect gas. In frozen and nonequilibrium solutions, the parameter \( \sigma \) as calculated from \( I(190) \) never approaches unity, even at high Mach numbers, because of the frozen-in dissociation energy in \( h_0 \). To represent the Mach number effect more realistically, the \( \sigma \) defined by \( I(190) \) is replaced by an effective value \( \sigma_e \), defined as

\[
\sigma_e = \left[ 1 + \frac{2}{(\gamma_e - 1) M^2} \right]^{-1}
\]

(37)

where \( \gamma_e \) is an effective specific heat ratio. For a perfect gas, (37) is equivalent to \( I(190) \). In BLAYER, \( \gamma_e \) is calculated as

\[
\gamma_e = \frac{a^2 W/R_0 T}{W}
\]

(38)

where \( W \) is the mixture molecular weight and \( a \) the sound speed.

The incompressible form factor \( HTR(I) \) is computed from \( I(189) \) and the displacement thickness \( DElBL(L) \) from \( I(188) \). In the case of a nonequilibrium flow solution, the coupling of the boundary layer to the inviscid flow is switched on when the indicator \( IUPD \) is set to zero in subroutine \( THRGAT \).* The only effect of the coupling in subroutine BLAYER is that the derivatives \( DDElBL(L) \) of the displacement thicknesses

*\( IUPD \) is always equal to 1 in the frozen and equilibrium solutions.
have to be calculated. To avoid excessive disturbance of the nonequilibrium solution, DDELBL(L) is gradually built up from zero over 29 steps in the flow solution, as explained in Section 7.6 of Volume I.

### 3.6.6 Heat Flux and Shear Stress

For FINAL = .FALSE., the boundary layer integral BLINT(L) is reset to its value BLINTI at the beginning of the step, and the boundary layer coordinate XI(L) to its initial value XII. Then control is transferred to statement 540, the limit of the Dσ loop over L.

For FINAL = .TRUE., the shear parameter XL is calculated from the curvefit I(194), the shear stress TAUW(L) from I(193), the Reynolds analogy factor RAF from I(203), and the heat flux QWDOT(L) from I(195). Then, if L is less than NPRFLS, control is transferred to the end of the Dσ loop over L. If L is equal to NPRFLS, several physical quantities are saved for use in the calculations for the next step. When the Dσ loop over L has been completed, diagnostic outputs are written if the control variable ISW4B is nonzero, and the RETURN is executed.

### 3.7 Subroutine BLCALL

During the nonequilibrium solution, all calls to the boundary layer routine BLAYER are made through subroutine BLCALL. Before calling BLAYER, subroutine BLCALL sets up certain data required by BLAYER and by the transport property routines. The species mole fractions are computed from equation I(1) and loaded into the common array SAVEC(J).

If the perturbation technique is still being used (INEQ = 0), the perturbations PERTG(J) are added to the concentrations \( γ_j = G(J) \) before equation I(1) is used. In this case, in addition, the temperature perturbation PCT is added to the nondimensional temperature CT and the density perturbation PRHØ is added to RHØ. After BLAYER has been called, the temperature and density are restored to their unperturbed (equilibrium) values. It is not necessary to reset SAVEC(J), because this array is used only in the transport, boundary layer, and model condition calculations, not in the inviscid
flow solution. Indeed, subroutine PRTA (which controls the output of the nonequilibrium flow solution) calls BLCALL to load the SAVEC array even in solutions neglecting the boundary layer (ISW3B = 0), because the output routine OUT1 obtains the mole fractions from SAVEC.

3.8 Subroutine BXSECT

Subroutine BXSECT is used in searching the array V in common block/TRANS7/for references to the indices of species pairs used in the cross section computations, and also in determining the correspondence between the entries in the V array and the steps in the cross section computations. The argument L of the subroutine is the index of the step in the cross section computations, MV is the index of the last entry in the V array used in step L of the computations, and I and J are the indices of the species pair referenced in V. When the subroutine is called with L = 0, the other arguments MV, I, and J are set internally by the subroutine and need not be furnished by the calling routine; however, all arguments are required when L ≠ 0.

In use, subroutine BXSECT is first called with L = 0 to begin the search over the V array. The routine then goes through each step L of the cross section computations, starting with L = 1 and checking for steps with KQ = 9 or 12. The value of MV is accumulated at each step L of the computations from the number of V values NV(KQ(L)) required for that step, where NV(KQ) is a fixed value which has been preset in the code. When a step with KQ = 9 or 12 is found, subroutine BXSECT sets I and J to the indices of the species pair referenced in the V array for that step and returns control to the calling program. The latter program may then reset I and J to any desired values and recall subroutine BXSECT again with the current values of L and MV to enter the revised data in the V array and resume the search of the computations again from the point at which it was left off. It should be noted that for each step with KQ = 12, there are two species pairs referenced in the V array, and returned by subroutine BXSECT, while for KQ = 9 there is only one.
When all steps of the computations have been searched, as indicated by the index $L$ of the step becoming equal to the number of steps $NKG$ in the edited cross section data in common block/TRANSY, the subroutine sets $L$ to 0, as a signal that the search has been completed, and returns control to the calling program.

3.9 Subroutine C\text{COMM}

The function of subroutine C\text{COMM} is to compute some physical quantities upon which the rates of change of the species concentrations and the other dependent variables depend. The actual calculations of the rates of change are then carried out in subroutine EXACT.

C\text{COMM} is called only from subroutine DERIVS, and the call to C\text{COMM} is always preceded by a call to subroutine THERM, which computes the species properties at the current nondimensional temperature, $CT = T/T_0$. In the case of an electronic nonequilibrium model, DERIVS calls THERM twice to obtain the required properties at both the gas temperature $CT$ and the electron temperature $CTE = T_e/T_0$.

To facilitate the discussion, the correspondence of some of the Fortran symbols with the mathematical notation used in Volume I is reviewed in the following table. Symbols not listed here are defined in Section 4.77.

<table>
<thead>
<tr>
<th>Fortran Symbol</th>
<th>Mathematical Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALGJ(J)</td>
<td>$\ln \gamma_j$</td>
</tr>
<tr>
<td>BETA(I,J)</td>
<td>$\beta_{ij}$</td>
</tr>
<tr>
<td>CHI(I)</td>
<td>$\chi_i$</td>
</tr>
<tr>
<td>CLKF</td>
<td>$\ln k_f$</td>
</tr>
<tr>
<td>CLKR</td>
<td>$\ln k_r$</td>
</tr>
<tr>
<td>CLNPI(I)</td>
<td>$\ln p_i$</td>
</tr>
<tr>
<td>CLNMC(I)</td>
<td>$\ln (1 - \chi_i)$</td>
</tr>
<tr>
<td>CLNTD(L)</td>
<td>$\ln T$ for $L = 1$; $\ln T_e$ for $L = 2$</td>
</tr>
</tbody>
</table>
\[ \ln \left( \rho \sum \gamma_j \right), \text{ sum over third bodies} \]

\[ \ln (R_0 T) \text{ or } \ln (R_0 T_e), \text{ argument in cm}^3 \text{ atm/mole} \]

\[ \ln (\rho R_0 T) \text{ or } \ln (\rho R_0 T_e), \text{ argument in atm-g/mole} \]

\[ T/T_0 \text{ for } L = 1; \ T_e/T_0 \text{ for } L = 2 \]

\[ \epsilon_{fi} \]

\[ \epsilon_{ri} \]

\[ \gamma_j \]

\[ p_i \]

\[ p_i \chi_i \]

\[ \dot{a}_e \]

\[ \dot{a}_r \]

\[ q_{fi} \]

\[ q_{ri} \]

\[ \rho \]

\[ \ln \rho \]

\[ \frac{1}{R_0} \sum_{j=2}^{n} \gamma_j \gamma_p \]

\[ u/u_s \]

\[ T_e \]

\[ \sum_{n=1}^{n} \nu_{ij} \ln \gamma_j \text{ or } \sum_{j=1}^{n} \nu_{ij} ' \ln \gamma_j \]
\[ \sum \gamma_j, \text{ sum over third-body species} \]

\[ T_8 = \sum_{j=1}^{n} \nu_{ij} \]

\[ T_9 = \sum_{j=1}^{n} \beta_{ij} \left[ \ln \left( \frac{R_0 T'}{R_0 T} \right) + \frac{\mu_j^0(T')}{R_0 T'} \right] \]

where \( T' \) is \( T \) or \( T_e \)

\[ XMJATD(J,L) = \frac{\mu_j^0(T')}{R_0 T'}, \text{ where } T' = T \text{ for } L = 1, \]
\[ T' = T_e \text{ for } L = 2 \]

\[ XNUIJ(I,J) = \nu_{ij} \]

\[ XNUIJP(I,J) = \nu_{ij} \]

\[ Z1 = \ln \left( \frac{1}{u} \right), \text{ for } u \text{ in cm/sec} \]

Subroutine \texttt{C@MM} may be divided, for purposes of explanation, into three parts: an initialization section, a main loop over the reactions, and a final section. These parts of the subroutine will now be discussed.

\subsection{3.9.1 Initialization Section}

This section extends from the subroutine entry point through the statement \texttt{RCALC = .FALSE.} (card \texttt{C@M 107}). It sets up the ALGJ array, computes a number of sums which are required later in \texttt{C@MM} or in other parts of NATA, calculates the nondimensional flow velocity \( SU \) from the energy equation \( I(245) \), and calls subroutine \texttt{GE@M} to compute the gas density.

In addition, it contains the controls for intermittent dumping of diagnostic data in the nonequilibrium routines. The indicator for these dumps is the integer variable \texttt{ISW5B} in blank common. \texttt{ISW5B} is preset to zero. For the zero
value, no dumps are produced. If ISW5B is set in the input to a positive value, then dumps are produced each time C0MM, EXACT, RNKT, or PRFA is called. These dumps provide a detailed trace of the operation of the nonequilibrium integration, but give several pages of printed output for each integration step. Thus, it is hardly feasible to run a complete solution with continuous dumps. However, operation of the nonequilibrium routines at late stages in the solution can be studied by calling for intermittent dumps. This is done by setting ISW5B to a negative value in the input. Upon the first entry into C0MM, the variable ICYCLE is then set from its preset value of 90000000 to ISW5B. The counter ICOUNT is incremented by 1 every time C0MM is entered. Whenever ICOUNT is an exact multiple of ICYCLE, ISW5B is reset to 1; otherwise, it is reset to 0. Thus, upon every ICYCLEth entry into C0MM, the dumps are switched on and are produced in all of the nonequilibrium routines until C0MM is entered the next time, when they are switched off again. The coding of these controls is in the first nine executable statements of C0MM.

3.9.2 Main Loop Over the Reactions

The main calculations in C0MM are performed in a large DO loop headed by the statement, DO 460 I = 1, ISR (card C0M 109). Figure 7 is a flowchart of this main loop. In essence, this loop performs only two functions:

1) First, it computes the quantities \( P_i \) and \( \chi_i \) appearing in the rate equations I(287).

2) Second, in the case of an electronic nonequilibrium model, it evaluates the energy transfer to the electron gas \( q_e \) and the radiative loss \( q_r \) due to the reactions.

The quantities \( P_i \) and \( \chi_i \) are computed from equations I(288) and I(289), respectively, in logarithmic form:

\[
\ln P_i = \ln \left( \frac{1}{u} \right) + \ln k_{f_i} + (\nu_i - 1) \ln \rho + \sum_{j=1}^{n} \nu_{ij} \ln \gamma_j \quad (39)
\]
Figure 7. Flow Chart of Main Loop Over Reactions in Subroutine GEM

ORIGINAL PAGE IS OF POOR QUALITY
In equation (40), the equilibrium constant $K_i$ in I(289) has been expressed in terms of more basic quantities using equation I(278). In the summations indicated in (39) and (40), terms in which the stoichiometric factor $V_{ij}$ or $P_{ij}$ is zero are omitted.

If $\ln(1 - \lambda_i)$ is less than 20, $P_i$ and $\lambda_i$ are computed from (39) and (40) using

$$P_i = \exp(\ln P_i) \quad (41)$$

$$\lambda_i = 1 - \exp[\ln(1 - \lambda_i)] \quad (42)$$

and $P_i \lambda_i$ is obtained by multiplication. On the other hand, if $\ln(1 - \lambda_i)$ is greater than or equal to 20, so that $1 - \lambda_i$ is greater than $4.85 \times 10^8$, the term 1 in $(1 - \lambda_i)$ is neglected and the product $P_i \lambda_i$ is computed as

$$P_i \lambda_i = -\exp[\ln P_i + \ln(1 - \lambda_i)] \quad (43)$$

This algorithm allows the computation of a finite $P_i \lambda_i$ value in situations where $P_i$ is so small that an exponential underflow occurs in (41), but $\lambda_i$ is negative and so large that $P_i \lambda_i$ is within the range representable by floating-point numbers. It also makes possible a simple treatment of the special cases which arise when one or more of the $\gamma_j$ are zero, as discussed below.

If a list of third-body (catalytic) species has been specified for the reaction, the expression I(288) for $P_i$ is replaced by

$$P_i = \left[ \frac{\rho \gamma_{i-1}}{u} k_{fi} \prod_{j=1}^{n} \gamma_{ij} \right] \cdot \rho \sum_{tb} \gamma_k \quad (44)$$
in which the sum on the right is over the third-body species specified. The corresponding logarithmic expression is (39) with a term \( \ln (\rho \sum \chi_k) \) added to the right. The third bodies do not affect \( \chi \). Equation (44) may be derived from I(283) and I(287) by noting that, for a catalytic species \( (k) \), \( \varphi_{ik} = \varphi'_{ik} = 1 \). Thus, a factor \( \chi_k \) can be removed from both terms on the right in I(283). If the \( i \)th reaction involves more than one third-body species, each such species gives a separate three-body reaction. The effects of these reactions can be summed to give a single reaction with an effective third-body concentration which is the sum of the concentrations of the actual catalytic species, as in (44).

In the case of an electronic nonequilibrium gas model, there are two temperatures, the gas temperature \( T \) and the electron temperature \( T_e \). The forward reaction rate \( k_{fi} \) can depend on either \( T \) or \( T_e \), or both, or neither. The form of temperature dependence of \( k_{fi} \) is specified by the array \( KTF(I) \). If the indicator \( KF = KTF(I) \) is equal to 1, \( k_{fi} \) is given by I(69). If \( KF \) is equal to 2, the temperature \( T \) in I(69) is replaced by \( T_e \). This procedure is coded by placing the two nondimensional temperatures into a two-element array \( CTD(L) \), and placing the logarithms of the two temperatures (expressed in degrees Kelvin) into an array \( CLND(L) \). Both cases are then treated by a single formula involving \( CTD(KF) \) and \( CLND(KF) \), i.e., statement 190 of C@MM, which evaluates \( \ln k_{fi} \) from

\[
\ln k_{fi} = \ln C_i + \gamma_i \ln T_k - \frac{E_{ai}}{R_0 T_k}
\]

where \( T_k \) is equal to \( T \) for \( k = 1 \) and equal to \( T_e \) for \( k = 2 \). In (45), \( \ln C_i = \ln A_i - \gamma_i \ln (10^4) \).

Three special forms for the forward reaction rate, used in the electronic nonequilibrium model for argon (Appendix A in Volume II), are indicated by \( KTF(I) = 3, 4, \) and 5. For \( KTF(I) = 3 \), \( k_f \) is assumed to be given by

\[
k_f = C_i T_e \left( 1 - e^{-E_i/R_0T_e} \right)
\]

where \( C_i \) and \( T_k \) are the two-dimensional temperatures.
where $E_i$ is a parameter with dimensions of energy. For $K_T(I) = 4$,

$$k_f = \frac{c_i T_e \gamma_i}{\max (1, \tau)}$$

(47)

where

$$\tau = b \rho \gamma_a R$$

(48)

Here $R$ is the local nozzle radius (or equivalent radius in the case of a channel), and $\gamma_a$ is the concentration of the product atom in the reaction. For $K_T(I) = 5$,

$$k_f = \frac{c_i}{\sqrt{R}}$$

(49)

In NATA, the reverse rate constant $k_{r_1}$ is calculated from the detailed-balancing relation I(277), using the equilibrium constant $K_i$, equation I(278). In electronic non-equilibrium gas models, the coding of C0MM allows for three types of temperature dependence of $K_i$. If the indicator $K_R = K_T(I)$ is zero, the reverse reaction is assumed to occur only at a negligible rate. If $K_R$ is 1, $K_i$ is calculated from I(278) using the gas temperature $T$. If $I_R = 2$, the temperature $T$ in I(278) is replaced by $T_e$. The chemical potentials $\mu_{j}^{0}$ in I(278) are temperature dependent. During calculations based on an electronic nonequilibrium model, both $\mu_{j}^{0}(T)$ and $\mu_{j}^{0}(T_e)$ are computed, and these data are stored in the doubly dimensioned array XMJAT(J,L) with dimensions (20,2). The coding of equation (40) utilizes the array with the index $L = K_R$ so that both cases can be treated using a single formula, card C0MM 153.

Cases in which one or more of the concentrations $\gamma_j = 0$ require some special consideration because of the appearance of $x_n \gamma_j$ in the formulas used in C0MM. In NATA, the $\gamma_j$ are represented by the double precision array GJ(J). A zero
value for an element of this array could come about in one of two ways:

(1) One of the concentrations GJ(J) could underflow as a result of its decrease during the flow solution. However, this is most unlikely to occur in practice because, to underflow, GJ(J) must fall below $10^{-75}$ (on the IBM 360/75) or below $10^{-308}$ (on the UNIVAC 1108). Since GJ(J) can decrease by no more than 10 percent in a single integration step because of step size controls, the number of steps required to reach underflow from initial reservoir values of (say) $10^{-1}$ to $10^{-20}$ is of the order of 1000 or more, even on the IBM 360. Also, unless the control parameter GAMIN were set to zero, the decrease of GJ(J) would be stopped, long before underflow was reached, by the freezing of minor species (Section 7.5.3 of Volume I).

(2) A zero GJ(J) value could arise accidentally, at an intermediate point in the Treanor-Runge-Kutta integration step, as a result of a negative $\Delta \gamma$ which happened to be exactly equal in magnitude to the $\gamma$ value at the start of the step. This is quite unlikely, but it could happen.

To allow for these rather remote possibilities, the coding of C0MM is designed to cope with zero concentrations if they should occur. If one of the species concentrations, $\gamma$, is zero, four special cases can arise, depending upon the values of the stoichiometric coefficients $\nu_{ik}$, $\nu'_{ik}$ in the reaction I(219):

Case 1: $\nu_{ik} = \nu'_{ik} = 0$  \hspace{1cm} (50a)

Case 2: $\nu_{ik} = 0$, $\nu'_{ik} > 0$  \hspace{1cm} (50b)
Case 3: $\nu_{ik} > 0, \nu_i'k = 0$  \hspace{1cm} (50c)

Case 4: $\nu_{ik} > 0, \nu'_ik > 0$  \hspace{1cm} (50d)

In the Cornell Aeronautical Laboratory program from which NATA was derived, each of these cases was treated separately by special coding. In the present version of the code, all of these cases are treated using the formulas (39) to (44) with the aid of a computational artifice.

Cases involving zero species concentrations $\gamma_j$ require special consideration because of the appearance of $\ln \gamma_j$ in the formulas (39) and (40). Any attempt of the program to evaluate $\gamma_j$ would lead to immediate termination of the run by the computer operating system. In subroutine C0MM, the quantities $\ln \gamma_j$ are stored in an array, ALGJ(J). For species whose concentrations $\gamma_j$ are zero, ALGJ(J) is set to a large negative number, $L$, whose magnitude is chosen in accordance with the following criteria:

1. $L$ must be sufficiently large in magnitude that, even when combined additively with other terms as indicated in (39) and (40), it is sufficiently negative to produce a guaranteed exponential underflow, i.e.,

$$\exp(L + \text{other terms}) = 0 \text{ (exactly)} \hspace{1cm} (51a)$$

This will happen so long as the argument of the exponential function is less than about -88 on the UNIVAC 1108 or less than about -180 on the IBM 360/75.

2. $L$ must be sufficiently small in magnitude that the operation

$$L + \text{other terms} - L = \text{other terms} \hspace{1cm} (51b)$$

gives a result for "other terms" containing several significant digits of accuracy. The "other
terms" in equations (39) and (40) are all natural logarithms of the values of physical quantities whose values lie roughly in the range $10^{-20}$ to $10^{20}$; thus, the logarithms range from about $-50$ to $+50$. The second condition can thus be satisfied by taking $L$ to be less than about $10^4$ or $10^5$. If $L$ were chosen to be much larger, say $10^{10}$, then the first addition in (1b) would give $L + \text{other term} = L$ (exactly) on the UNIVAC 1108, because the change in $L$ due to addition of the "other terms" would be less than the accuracy with which $L$ can be represented as a floating-point number (about 8 significant figures).

In subroutine $C\text{OMM}$, $L$ has been chosen to be $-1.00$, a value which clearly satisfies both conditions.

Now assume that a particular concentration $\gamma_k$ is zero. With $\ln \gamma_k = L$, the formulas (39) and (40) treat the four cases (50) as follows:

In Case 1, $\nu_{ik}$ and $\beta_{ik}$ are both zero, so that the term $\ln \gamma_k$ does not actually appear in either (39) or (40). In this case, according to (50a), the species $k$ whose concentration $\gamma_k$ is zero does not participate in the reaction as either a reactant or a product.

In Case 2, $\ln \gamma_k$ does not appear in (39), so that the evaluation of $\ln P_i$ proceeds normally. However, $\beta_{ik} = \nu'_{ik}$ is positive, so that the right hand side of (40) contains a term $\nu'_{ik} L$ which is negative and large in magnitude. The resulting exponential underflow during the evaluation of (42) then gives $X = 1$. This is the correct result for this case, since $\gamma_k^{\beta_{ik}}$ is a zero factor in the second term on the right in equation I(289).

In Case 3, equations (39) and (40) can be written

$$\ln P_i = \ln \left( \frac{1}{\mathcal{U}} \right) + \ln \kappa_i + (\nu_i - 1) \ln \rho + \sum_{j \neq k} \nu_{ij} \ln \gamma_j + \nu_{ik} L$$

(52a)
\[ \ln (1 - \chi_i) = \beta_i \ln (\rho R_0 T) + \sum_{j \neq k} \beta_{ij} \left[ \ln \gamma_j + \frac{\mu_i^{0}}{R_0 T} \right] \]

- \gamma_{ik} \left( L + \frac{\mu_k^{0}}{R_0 T} \right) \quad (52b)

Here the term \(- \gamma_{ik} L\) on the right in (52b) is positive and very large; thus, \(i_k\) equation (42) is not used and the product \(P_i \chi_i\) is evaluated using (43), which gives

\[ P_i \chi_i = - \exp \left\{ \ln \left( \frac{1}{u} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \beta_i \ln (\rho R_0 T) \right. \]

\[ + \sum_{j \neq k} \nu'_{ij} \ln \gamma_j + \sum_{j = 1}^{n} \beta_{ij} \frac{\mu_i^{0}}{R_0 T} \right\} \quad (53) \]

\[ = - \frac{k_{fi} \rho^{\nu_i - 1}}{k_i} \prod_{j \neq k} \gamma_j^{\nu'_{ij}} \]

The terms involving \(L\) have cancelled out. The validity of equation (53) for Case 3 can be verified directly by multiplying equation (288) and (289), letting \(\gamma_k \to 0\), and taking (50c) into account.

In Case 4, equations (39) and (40) give

\[ \ln p_i = \ln \left( \frac{1}{u} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \sum_{j \neq k} \nu_{ij} \ln \gamma_j \]

\[ + \nu_{ik} L \quad (54a) \]

\[ \ln (1 - \chi_i) = \beta_i \ln (\rho R_0 T) + \sum_{j \neq i} \nu_{ij} \left[ \ln \gamma_j + \frac{\mu_i^{0}}{R_0 T} \right] \]

\[ + (\nu'_{ik} - \nu_{ik}) \left[ L + \frac{\mu_i^{0}}{R_0 T} \right] \quad (54b) \]

-57-
Addition of these two equations gives

\[
\ln [p_i (1 - x_i)] = \ln (\frac{1}{u}) + \sum k_{fi} + (\nu_i - 1) \ln \rho + \beta_i \ln (\rho R_0 T) \\
+ \sum_{j \neq k} \nu_{ij} \ln y_j + \sum_{j \neq k} \beta_{ij} \frac{\mu_{ij}}{R_0 T} + \beta_{ik} \frac{\mu_{k0}}{R_0 T} \\
+ \nu_{ik} L
\]  

(55)

Since \( \nu_{ik} L \) is negative and large in magnitude, (55) implies that

\[
p_i (1 - x_i) = p_i - p_i x_i = 0
\]  

(56)

Also, (54a) and (41) give \( p_i = 0 \); thus, \( p_i x_i = 0 \). This result is correct for Case 4, as may be seen by setting one of the \( \gamma_k \) to zero in (286) and taking (50d) into account.

The preceding analysis can easily be generalized to include cases in which more than one of the concentrations is zero. Thus, the artifice of setting \( \ln \gamma_k = L \) when \( \gamma_k = 0 \) provides a satisfactory treatment of all cases that can arise.

In calculations for a chemical nonequilibrium model (\( NT = 1 \)), the main loop provides only calculations of \( p_i \) and \( p_i x_i \). In calculations for an electronic nonequilibrium model, the loop also computes the contributions to the energy transfer to the electron gas (QDPE) and the radiative power loss (QDPR). These computations are provided by the coding below statement 390, and simply consist of evaluation of the formulas

\[
(\Delta q_i) = k_{fi} \varepsilon_f \prod_{j=1}^{n} (\rho \gamma_j)^{\nu_{ij}} - k_{ri} \varepsilon_r \prod_{j=1}^{n} (\rho \gamma_j)^{\nu'_{ij}}
\]  

(57a)
which are obtained by combining equations I(320) and I(321b,c). In equations (57), \( \Delta \dot{q}_i \) and \( \Delta \dot{r}_i \) denote the contributions of the \( i \)th reaction to \( \dot{q}_i \) and \( \dot{r}_i \), respectively. The energy partition parameters \( \epsilon_{fi}, \epsilon_{ri}, q_{fi}, q_{ri} \) are obtained by calling subroutine EPART. Effects of third-body species, if any, are taken into account using the quantity \( \text{CLTBF} = \frac{f_{tb}}{P_i \gamma_i} \), already evaluated during the computation of \( P_i \gamma_i \).

\[ (\Delta \dot{q}_i)_i = k_{fi} q_{fi} \prod_{j=1}^{n} (\rho \gamma_j)^{\nu_{ij}} - k_{ri} q_{ri} \prod_{j=1}^{n} (\rho \gamma_j)^{\nu'_{ij}} \]  

(57b)

3.9.3 Final Section

After the main loop over the reactions has been completed, \( \text{CMM} \) checks the value of \( \text{INEQ} \). If this indicator is zero (i.e., if the perturbation technique is still being used to generate the flow solution), the \( \text{RETURN} \) is executed immediately. If \( \text{INEQ} \) is nonzero (i.e., if the numerical integration of the rate equations has been started), \( \text{CMM} \) calculates the non-dimensional pressure, \( \text{PRES} = \frac{p}{p_0} \). If a chemical nonequilibrium model is being used, the pressure is computed from equation I(273) in the form

\[ \frac{p}{p_0} = \frac{\rho}{\rho_0} \frac{T}{T_0} \frac{W_0}{W} \]  

(58)

If an electronic nonequilibrium model is in use, equation I(323) is used to calculate the pressure in the form

\[ \frac{p}{p_0} = \frac{\rho}{\rho_0} \frac{\gamma e^{\gamma T_0} + \gamma h T}{T_0} \frac{W_0}{W} \]  

(59)

In either case, \( \frac{p}{p_0} \) is then divided by \( \text{RBOBARA} = \frac{\rho}{\rho_0} \) to eliminate the effects of gas imperfections, if any, upon the value of the reservoir density, \( \rho_0 \). This is done because gas imperfections are not taken into account in the nonequilibrium integration, as explained in Section 2.5 of Volume I. In addition, the gas entropy is calculated from equation I(251).
If an electronic nonequilibrium model is in use, CÔMM computes the contribution to the electron energy transfer \( \dot{q}_e \) from elastic collisions, using the formula

\[
( \Delta \dot{q}_e )_{\text{elas}} = \sum_{j=2}^{n} \frac{2W_e}{W_j} \cdot \frac{3}{2} \frac{R_0(T - T_e)}{N_0} \frac{n_e}{n_j} \frac{\nu_{ej}}{\nu_{eij}}
\]

(60)

where for neutral species \( j \), the collision frequency \( \nu_{ej} \) is given by

\[
\nu_{ej} = \frac{4}{3} \sqrt{\frac{8 R_0 T_e}{\pi W_e}} n_j Q_{ej}^{-1/(1,1)}
\]

(61)

and for ionic species

\[
\nu_{ei} = \frac{8}{3} \sqrt{\frac{2}{n_e}} \frac{n_{ie}^4}{(2k T_e)^{3/2}} \frac{\ln \left( \frac{k T_e^2}{\pi n_e^2} \right)}{(2k T_e)^{3/2}}
\]

(62)

These formulas are derived in Appendix A of Volume II; see equations II(25-31). The notation is

- \( W_e, W_j \) molecular weights for electron and \( j \)th species
- \( n_e = N_0 \rho \gamma_e \) electron concentration
- \( n_j \) concentration of a neutral atom or molecule
- \( n_i \) concentration of an ion
- \( Q_{ej}^{-1/(1,1)} \) Maxwell-averaged momentum transfer cross section (assumed the same for all neutrals)
- \( m_e \) electron mass
- \( k \) Boltzmann constant
- \( N_0 \) Avogadro number
- \( e \) electronic charge (esu)
In NATA, the cross section $Q_{ej}^{(1,1)}$ is obtained by calling the function PIØMEG(TEP). The constant factors in (61) and (62) are represented by VJC1, VJC2, and VNC, and are preset in a data statement in CØMM. Equation (60) is evaluated straightforwardly, and the result $Q_{elas} = (\Delta q_e)_{elas}$ is added to the $q_e$ due to the reactions to obtain the total rate of energy transfer to the electron gas.

3.10 Subroutine CXSECT

Subroutine CXSECT is used by subroutine XSECT to determine the correspondence between species pairs in the master species list for the code and in the rearranged species list stored in the common variable I(J) (see the discussion of subroutine XSECT below). When the subroutine is called with its arguments set equal to the indices of a pair of species in the master species list, it returns the arguments as the indices of the corresponding species pair in the rearranged list. If the original indices are denoted by L,J, the corresponding revised indices are I(L), I(J) if L-J and I(L)-I(J) have the same sign, and are I(J), I(L) otherwise.

3.11 Subroutine DERIVS

During the nonequilibrium solution, the derivatives of the dependent variables, including the species concentrations, are calculated by calling subroutines THERM, CØMM, and EXACT in sequence. If an electronic nonequilibrium gas model is in use, THERM must be called twice, once with $CT = T/T_0$, and once with $CT = T_e/T_0$, to provide species properties at both the gas temperature and the electron temperature. The use of these properties has been explained in the analysis of subroutine CØMM. Also, if the boundary layer is being included in the flow solution, an iteration is necessary to determine the self-consistent solution for the boundary layer displacement thickness $\delta$ and the derivatives of the flow variables (Section 7.6 of Volume I). These calculations are called for at several locations in subroutines NØNEQ and RNKT. For compactness of coding, these operations are all placed in a separate subroutine, DERIVS.
For an electronic nonequilibrium model (TN = 2), DERIVS saves the current value of CT in the location CTSAVE, resets CT to CTE = T_e/T_0, and calls THERM to compute the species properties at the electron temperature. The values \( \mu_i^0(T_e)/R_0T_e \) are stored directly into XMJATD(I,2). The enthalpy, specific heat, and entropy of the electrons are saved temporarily in the locations HJE, CPJE, and SENTE. Then CT is reset to the value T/T_0 and THERM is called again to compute the species properties at the gas temperature. THERM overwrites the previously calculated data for the electrons (species 1), but these are reset from HJE, CPJE, and SENTE.

For a chemical nonequilibrium model (NT = 1), THERM is called only once, to obtain the properties for all the species at the gas temperature, T.

In either case, DERIVS next calls CPMM and EXACT to compute the derivatives of the species concentrations and other dependent variables. It then applies the condition I(407b); if this condition is violated, the indicator FAILED is set to .TRUE. and a RETURN is executed.

The remaining portion of DERIVS controls the iteration to obtain a self-consistent solution for the boundary layer displacement thickness and the derivative \( d \ln M/dx \) of the logarithm of the Mach number. If the boundary layer is not being included in the flow solution (ISW3B = 0), or if the solution is still in the upstream (uncoupled) region (IUPD = 1), this iteration is not performed. The technique used in the iteration is explained in Section 7.6 of Volume I, in the vicinity of equation I(424).

### 3.12 Subroutine DSMSØL

DSMSØL is the subroutine called for simultaneous solution of systems of linear equations in the calculation of thermochemical equilibrium conditions, in the equilibrium flow solution, and in the nonequilibrium solution. The calling sequence for DSMSØL was embodied in the original program version received from Cornell Aeronautical Laboratory (reference 6), but the subroutine itself was not provided by CAL because it was written in machine language and would not be operable on other types of computer system. To provide the required
capability for solution of systems of equations, subroutine SIMQ from the IBM Scientific Subroutine Package (reference 7) was incorporated into NATA, and a Fortran DSMSGL routine was written to rearrange the data for the equations from the form provided by the CAL routines into the form required by SIMQ.

For a system of linear equations

$$\sum_{k=1}^{i} a_{jk}x_k = b_j \quad (j = 1, \ldots, i) \quad (63)$$

in i unknowns \((x_1, \ldots, x_i)\), NATA stores the coefficients \(a_{jk}\) and constants \(b_j\) in the double precision array \(AA\) in unlabelled common. The dimensions of \(AA\) are \((22, 24)\). The matrix of coefficients \(a_{jk}\) is stored as

$$a_{jk} = AA(j, k) \quad \begin{cases} j = 1, \ldots, i \\ k = 1, \ldots, i \end{cases} \quad (64)$$

The constants \(b_j\) are stored as

$$b_j = AA(j, i + 1) \quad (65)$$

Subroutine SIMQ (discussed below) requires that the coefficients \(a_{jk}\) be stored columnwise in a singly dimensioned array \(A\), without gaps, and that the constants \(b_j\) be stored in an array \(B\). DSMSGL reorganizes the data into this form, using the storage locations of the array \(AA\) to contain \(A\) and \(B\). The array \(A\) is assigned to the first \(i^2\) storage locations in \(AA\), that is, \(AA(1,1), AA(2,1), \ldots, A(22,1), AA(1,2), AA(2,2), \ldots\)

The array \(B\) is assigned the first \(i\) storage locations in the 23rd column of \(AA\), that is, \(AA(1,23), \ldots, AA(i,23)\). If the number of equations \(i\) is equal to 22, the reorganization leaves the matrix \(AA\) unaltered. If \(i\) is less than 22, the coefficients \(a_{jk}\) are shifted down so as to fill up the first columns of \(AA\) without gaps, and the constants \(b_j\) are shifted
up into the previously unused 23rd column. After SIMQ has been called, the solution values $x_k$ provided by SIMQ are shifted from the 23rd column into the $(i+1)$th column, for use by the calling routine.

The coding of DSMSOL is straightforward, and can be followed easily with the help of the definitions given in the glossary of Fortran symbols (Section 4.80).

3.13 Subroutine DUMP

DUMP is part of the NATA system for dealing with possible errors in code operation. NATA contains numerous validity checks. If one of these is violated, the routine in which the error was detected calls DUMP. This small subroutine sets the indicator ERR to .TRUE. and writes a message giving the name RNAME of the calling routine. It then returns control to the calling routine. The calling routine then executes a RETURN. At each higher-level routine, statements of the form IF(ERP) RETURN pass control up still higher levels until the main program is reached. In MAIN, when a .TRUE. value of ERR is detected, subroutine DUMPEX is called to print diagnostic data (see below). After the return from DUMPEX, MAIN proceeds as if a normal case completion had occurred; i.e., the job is not terminated unless the input ISW4A is zero.

Originally, the operations now performed by DUMP and DUMPEX were included in a single subroutine (called DUMP). When NATA become so large that it had to be run using overlay on the UNIVAC 1108, the original DUMP routine was divided into two parts, and the larger of these (DUMPEX) was placed into a separate overlay segment so as to minimize the core storage requirement.

3.14 Subroutine DUMPEX

DUMPEX produces the diagnostic dumps when an error has been detected in the operation of the code. The routine contains unlabelled common and many of the labelled common blocks in NATA. When DUMPEX is executed, most of the variables and arrays listed in these common statements are printed out
in namelist format. In addition, if a binary tape of data for plotting is being written (DATAPE = .TRUE.), DUMPEX backspaces this tape to eliminate all of the records written during the current case.

3.15 Subroutine ELC\(\text{AND}\)

Subroutine ELC\(\text{AND}\) computes the electrical conductivity SIGMA of the gas in mhos/cm from equation I(98) using the values of SSIG = (e/k)^2/\(\sqrt{T}\) and \(\Omega(1,1,J) = k^{-1}\sqrt{T} \Delta[1]_{i}\) computed in subroutine TRANSP. In this computation it is assumed that if electrons are present in the gas, they are the first species. If, on the other hand, electrons are not present (IELEC = 0), the electrical conductivity is set to 0.0 and the computation is bypassed.

3.16 Subroutine ELTIME

The function of subroutine ELTIME is to determine and print out the elapsed execution time since the beginning of the run. When NATA is run on the UNIVAC 1108, ELTIME uses the utility routines .SET and TIME, which are provided by the EXEC II operating system. When the code is used on the IBM 360/75 at Avco, the functions of RESET and TIME are simulated by a small Fortran program (called RESET, Sec. 3.58) which utilizes the system subroutine ACUCPU.

ELTIME contains a logical indicator CALLED, which is preset to .FALSE. in a data statement. When ELTIME is called for the first time in a job, a test on CALLED leads to a call to RESET, which initializes the time counter I for subroutine TIME to 0, and ET\(\text{O}\) and ET are both set to zero. Also, CALLED is set to .TRUE., so that this part of the routine is bypassed when ELTIME is called on subsequent occasions in the run.

In such subsequent calls, ELTIME sets ETP = ET\(\text{O}\), and then sets ET\(\text{O}\) to ET, where ET is the value left in storage by the previous call (unless this argument has been altered by the calling routine). Then ELTIME calls subroutine TIME(I), which returns the elapsed time I since the call to RESET, measured in milliseconds. ELTIME converts this time to seconds

-65-
and stores it in ET. Then, unless the argument IP is zero, ELTIME converts ET into minutes and stores the result in ETM, and also computes the elapsed time DET in seconds since the last previous printed output of the time. A message is then printed giving ETM and DET. If the printed message is skipped because IP = 0, ETØ is reset to ETP to maintain its significance as the elapsed time since the printing of the last time message.

3.17 Subroutine EPART

This subroutine computes the parameters governing the partition of reaction energy between radiative loss and energy transfer to the electron gas for each reaction in a gas model which includes electronic nonequilibrium. The arguments in the calling sequence are defined in Section 4.85. The subroutine treats six cases, or types of reactions, which are distinguished by an index IT. In the following discussion of these cases, \( \varepsilon_f = EF \) and \( -\varepsilon_r = -ER \) denote the energies gained by the electron gas in one mole of reactions in the forward and reverse directions, respectively; and \( q_f = QF \) and \( -q_r = -QR \) denote the corresponding energies lost by radiation.

The six cases are

\[ IT = 1 \quad \varepsilon_f = -a R_0 T_e \]
\[ q_f = \varepsilon_0 - \varepsilon_f \]
\[ \varepsilon_r = q_r = 0 \]

\[ IT = 2 \quad \varepsilon_f = -\frac{3}{2} R_0 T_e \]
\[ q_f = \varepsilon_0 - \varepsilon_f \]
\[ \varepsilon_r = q_r = 0 \]
In these formulas, $\varepsilon = E_0$ denotes the energy available for partitioning between the electrons and the radiative losses, $R$ the universal gas constant ($1.9872 \text{ cal/mole-deg}$), and $T_e$ the electron temperature ($^\circ\text{K}$).

3.18 Subroutine EQCALC

EQCALC computes the conditions of thermochemical equilibrium at specified temperature $T$ ($^\circ\text{K}$) and pressure $P$ (atm), using the method explained in Section 6.1 of Volume I. As an aid to following the coding, the correspondence of Fortran and mathematical symbols is reviewed in the following list:

- $\text{BET}(I)$, $\nu^{*} - 1$ (I = i-c)
- $\text{CDIJ}(I,J)$, $\nu^{*}$ (I = i-c, j)
- $\text{CGI}(I)$, $W_i$
- $\text{QM}(I)$, $q_i$

\[ IT = 3 \quad \varepsilon_f = \varepsilon_r = q_f = q_r = 0 \]

\[ IT = 4 \quad \varepsilon_f = \varepsilon_r = -\frac{3}{2} RT_e \]
\[ q_f = q_r = 0 \]

\[ IT = 5 \quad \varepsilon_f = \varepsilon_r = \varepsilon_0 \]
\[ q_f = q_r = 0 \]

\[ IT = 6 \quad q_f = \varepsilon_0 \]
\[ \varepsilon_f = \varepsilon_r = q_r = 0 \]
EQCALC first saves the current value of the nondimensional temperature CT in the location CTSAVE. This is done because EQCALC may be called with temperatures T differing from the temperature at the current flow point (for example, during the equilibrium normal shock solution). The nondimensional temperature CT is then reset to \( T/T_0 \) to allow calculation of the species thermal properties by calling subroutine THERM.

Next, EQCALC computes initial estimates of the species mole fractions and related quantities. The mole fractions for the independent species \( (I = 1 \text{ to } ISC) \) are set to \( q_i = QM(I) \), defined by equation I(10). If any of the \( QM(I) \) are zero or negative (due to roundoff errors), the corresponding ZCAP(I) values are reset to \( 10^{-3} \). Subroutine THERM is then called to compute the thermal properties of all the species at temperature \( T \). The mole fractions for the dependent species \( (I = ISC + 1 \text{ to } ISS) \) are then estimated using equation I(224), which can be written in the logarithmic form

\[
\ln X_i = CHII(i-c) + \sum_{j=1}^{C} \bar{\nu}_{i-c,j} \ln X_j
\]  

(66)

where

\[
CHII(i-c) = -\frac{\mu_i^o}{R_0 T} + (\nu_i^* - 1) \ln p + \sum_{j=1}^{C} \bar{\nu}_{i-c,j} \frac{\mu_j^o}{R_0 T}
\]  

(67)

In this calculation, SKIL(i) is used as an intermediate variable, first to represent \( \ln X_{i-c} \), then \( X_{i-c} \). If the number of dependent species, ISMC = n-c, is zero, then the calculation of
the dependent species mole fractions is skipped here and else-
where in EQCALC.

Once initial estimates of the \(X_i\) and \(\text{in} X_i\) have thus been
determined, EQCALC enters the Newton-Raphson iteration based on
equations I(277) to I(233). The iterative loop runs from state-
ment 70 down to (but not including) statement 220. The constant
terms \(F_j\), equation I(227), in the system of linear equations
I(233a) for the corrections \(h^n_r\) are loaded into the \(M^{\text{th}}\) column
of the matrix AA by the DØ loop ending at statement 90. Note
that, in statement 80, \(X_i\) for \(i = c + 1\) to \(n\) is denoted by
SKIL(i). Then the matrix of coefficients \(-X^n_r(\partial F_j/\partial X^n_r)\), equation
I(233b), is loaded into the first ISC rows and columns of
AA by the nest of DØ loops ending at statement 140. Subroutine
DSMSØL is then called to solve the system of equations for the
correction factors \(h^n_r\). Upon the return from DSMSØL, the solution
values for these quantities are obtained from the \(M^{\text{th}}\) col-
umn of the matrix AA, where \(M = ISC + 1\).

The corrected values of the species mole fractions are then
calculated. The mole fractions for the independent species
\((i = 1\) to \(c)\) are obtained using equation I(234):

\[
X^{n+1}_k = (1 + h^n_k) \cdot X^n_k
\]

(68)

The factor \(1 + h^n_k\) is denoted by ZA. If ZA is zero or negative,
the corrected mole fraction is estimated by halving the previous
value:

\[
X^{n+1}_k = X^n_k / 2
\]

(69)

These calculations are done in the DØ loop ending at statement
180.

The corrected mole fractions for the dependent species are
calculated from the equilibrium relations I(224), based on the
corrected \(X_i\) for the independent species. Equations (66) and
(67) are again used, in the DØ loop ending at statement 200.

The convergence test for the Newton-Raphson iteration is
that the absolute values of all the \(h^n_k\) be less than or equal
to a criterion value TEST, which is preset (in subroutine INIT)
to \(10^{-6}\). If convergence has not been achieved after \(NTEST = 100\)
iterations, the DUM routine is called to terminate the case and
produce diagnostic output.
When convergence has been achieved, EQCALC computes a number of gas properties based on the temperature \( T \), the pressure \( P \), and the equilibrium mole fractions determined by the Newton-Raphson solution of equations I(227). The nondimensional molar entropy of the gas mixture, \( ZSEN = \frac{S}{R_0} \), is calculated using equation I(30b). Note that the specific entropy of the gas is equal to

\[
s = \left( \frac{S}{R_0} \right) \frac{R_0}{W} = \frac{ZSEN \cdot CRA}{ZCM}
\]

in \( \text{cal/gm} \text{O}_\text{K} \). The nondimensional molar enthalpy \( ZCH \) of the gas is calculated from equation I(238) and the molecular weight \( ZCM \) from equation I(236). Finally, the corrections for gas imperfections, equations I(79) and I(81), are applied.

### 3.19 Subroutine EXACT

This subroutine computes the derivatives of the species concentrations and the other dependent flow variables for use in the nonequilibrium solution. EXACT consists of two sections. The first section (down to statement 100) calculates the derivatives required in the perturbation solution; the second gives those used in the nonequilibrium integration.

#### 3.19.1 Derivatives for the Perturbation Solution

The derivatives required in the perturbation solution (Section 7.3 of Vol. I) are those of the equilibrium flow variables. These derivatives are computed in EXACT by solving a system of \( n+2 \) linear equations of the form:

\[
\sum_{j=1}^{n+2} A_{ij} X_j = B_i
\]

where

\[
X_j = \frac{d \bar{v}_j}{dx} \quad \text{for } j = 1 \text{ to } n
\]
\[ x_{n+1} = \frac{1}{T_0} \frac{d \overline{T}}{dx} \]  
(72b)

\[ x_{n+2} = -\frac{d \ln \overline{\rho}}{dx} \]  
(72c)

The system of equations is

\[ i = 1 \text{ to } c:\quad \sum_{j=1}^{n} \alpha_{ji} \frac{d \overline{\gamma}_j}{dx} = 0 \]  
(73a)

\[ i = c+1 \text{ to } n:\quad \sum_{j=1}^{c} \frac{\overline{\nu}_{i-c}}{\overline{\gamma}_i} \frac{d \overline{\gamma}_i}{dx} + \frac{1}{\overline{\gamma}_i} \frac{d \overline{\nu}_i}{dx} + \frac{T_0}{\overline{T}} \left[ \frac{\overline{H}_i/\overline{T}_0}{\overline{T}/T_0} - 1 - \sum_{j=1}^{c} \overline{\nu}_{i-c,j} \left( \frac{\overline{H}_i/\overline{T}_0}{\overline{T}/T_0} - 1 \right) \right] \frac{d (\overline{T}/T_0)}{dx} + (\overline{\nu}_{i-c} - 1) \cdot \frac{d \ln \overline{\rho}}{dx} = 0 \]  
(73b)

\[ i = n+1:\quad \sum_{j=1}^{n} \frac{d \overline{\gamma}_j}{dx} + \frac{T_0}{\overline{T}} \left[ \sum_{j=1}^{n} \overline{\gamma}_j \right] \frac{d (\overline{T}/T_0)}{dx} + \left[ \frac{1}{\overline{W}} - \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{T_0}{\overline{T}} \right] \frac{d \ln \overline{\rho}}{dx} = \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{T_0}{\overline{T}} \cdot \frac{d \ln \overline{A}}{dx} \]  
(73c)

\[ i = n+2:\quad \sum_{j=1}^{n} \frac{\overline{H}_j}{R_0 T_0} \frac{d \overline{\gamma}_j}{dx} + \left[ \sum_{j=1}^{n} \overline{\gamma}_j \frac{\overline{\rho}_{pj}}{R_0} \right] \frac{d (\overline{T}/T_0)}{dx} - \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \cdot \frac{d \ln \overline{\rho}}{dx} = \frac{u^2}{u_s^2} \cdot \frac{u_s^2}{p_0 T_0} \cdot \frac{d \ln \overline{A}}{dx} \]  
(73d)
Equation (73a) is I(363) with the meanings of the indices interchanged; equation (73b) is I(367) with the signs of all terms reversed and the index replacements $j \rightarrow i, f \rightarrow j$; equation (73c) is I(299) applied to the equilibrium flow; and (73d) is I(293) applied to the equilibrium flow. Note that

\[
\frac{H_i}{R_0 T_0} = \text{SHJ}(i) \quad (74a)
\]

\[
\frac{F}{T_0} = C T 
\]

\[
\frac{u_s^2}{u_s} = SU2 \quad (74c)
\]

\[
\frac{u_s^2}{R_0 T_0} = \frac{R_0 T_0}{W_0} \cdot \frac{1}{R_0 T_0} = \frac{1}{W_0} \quad (74d)
\]

Equation I(296) is used in the coding of the quantity $1/W$ in (73c).

The coefficients $A_{ij}$ for these equations (73) are loaded into the array $AAA(I, J)$ for $I = 1$ to $n + 2$ and $J = 1$ to $n + 2$. The constants $B_i$ are loaded into $AAA(I, n + 3)$ for $I = 1$ to $n + 2$. The system of equations is then solved by calling subroutine $DL$, and the resulting values for the derivatives (72) are retrieved from the $(n + 3)$th column of $AAA$.

3.19.2 Derivatives for the Numerical Integration

During the numerical integration of the rate equations, the derivatives of the species concentrations are given explicitly by equation I(287). These derivatives are computed in the nest of $D\bar{O}$ loops ending at statement 120; they are denoted by $DGJ(J)$. 

- 72 -
The nature of the remaining derivatives to be determined depends upon the region of solution (upstream or downstream) and the type of gas model being used. The unknowns solved for in the various cases are summarized in Table I. When an electronic nonequilibrium model is being used (NT=2), there is a fourth dependent variable in the integration, the total enthalpy h₀. However, its derivative is given explicitly by equation I(333), so that it is not involved in the simultaneous solution for the derivatives listed in Table I.

For a given type of gas model, the same set of equations is solved regardless of whether the solution is in the upstream region (inverse method) or the downstream region (direct integration). However, the matrix of coefficients and the vector of constants differ in the two cases because of the difference between the sets of unknowns being solved for. Figure 8 is a flowchart of the solution for the derivatives. By following this flowchart, one can easily write out the equations being solved in each of the four cases listed in Table I, and verify that they are identical with the text equations listed in the table. It should be noted that NE = NT + 1 is the number of unknowns and equations in each case. For NT = 2, the variable SCPGH contains the sum \( \sum_{j=2}^{B} C_p j \gamma_j \) but for NT = 1 the sum is over all species (see subroutine COOL).

3.20 Function EXP

The operating system on the IBM 360/75 at Avco produces several lines of diagnostic output whenever the Fortran function DEXP(X) underflows to give a value 0. To avoid such interruptions of the normal output, the IBM 360 version of NATA contains a Fortran-coded exponential function, EXP(X), which checks the argument X and sets the result to zero for \( X \leq -180 \). For \( X > -180 \), EXP calls DEXP to compute EXP. This routine is not used in the UNIVAC version of NATA because the UNIVAC system EXEC II does not produce a diagnostic message for exponential underflow.
### TABLE I
DERIVATIVES DETERMINED BY EXACT DURING THE NUMERICAL INTEGRATION

<table>
<thead>
<tr>
<th>Region</th>
<th>IUPD</th>
<th>Model Type</th>
<th>NT</th>
<th>Unknown Derivatives</th>
<th>Equations Solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upstream</td>
<td>1</td>
<td>Chemical</td>
<td>1</td>
<td>( \frac{dT}{T_0} ), ( \frac{d\ln A}{dx} )</td>
<td>I(293), I(299)</td>
</tr>
<tr>
<td>Upstream</td>
<td>1</td>
<td>Electronic</td>
<td>2</td>
<td>( \frac{dT}{T_0} ), ( \frac{d\ln A}{dx} ), ( \frac{dT_e}{T_0} )</td>
<td>I(328), I(331), I(332)</td>
</tr>
<tr>
<td>Downstream</td>
<td>0</td>
<td>Chemical</td>
<td>1</td>
<td>( \frac{dT}{T_0} ), ( \frac{d\ln \rho}{dx} )</td>
<td>I(293), I(299)</td>
</tr>
<tr>
<td>Downstream</td>
<td>0</td>
<td>Electronic</td>
<td>2</td>
<td>( \frac{dT}{T_0} ), ( \frac{d\ln \rho}{dx} ), ( \frac{dT_e}{T_0} )</td>
<td>I(328), I(331), I(332)</td>
</tr>
</tbody>
</table>
Figure 8. Flowchart of the Solution for the Derivatives of the Flow Variables in the Numerical Integration.
3.21 **Subroutine FINDX**

The purpose of this subroutine is to determine the value of $x$ at which the geometric area ratio has a given value $A$. An indicator (UPDOWN) specifies whether the upstream or the downstream solution is desired. An entry point FINDXC(A,MBL,X) permits determining the value of $x$ at which a rectangular channel has a half-width $A$, in centimeters, on its MBLth profile.

The solution is obtained by a Newton-Raphson iterative procedure. The independent variable in the iteration is taken to be $V = |x|$.

If the area ratio is given (normal entry point), then its value is tested before the solution is attempted. If $A$ is less than 0.99999, the case is terminated by calling DUMP, because by its definition the area ratio cannot be less than unity. If $A$ is between 0.99999 and 1.0, the discrepancy is assumed to be due to a numerical (e.g., round-off) error elsewhere in the program; in this case, $X$ is set to 0.0 (the value for $A = 1$), and the RETURN is executed.

If $A$ is greater than 1.0, $V$ is initialized to 1.0 and the Newton-Raphson procedure is started. The geometric area ratio $AR$ and its derivative $DADX$ at the point $X = V \cdot \text{sign}$ (UPDOWN) are calculated by calling GEOMAR. If $AR$ agrees with $A$ to within one part in $10^5$, the iteration has converged. If not, a new estimate of $V$ is calculated from the Newton-Raphson formula

$$V = V_0 + \frac{A-AR}{|DADX|}$$

However, $V$ is not allowed to increase by more than a factor of 2 in any step; this is to prevent $V$ from becoming very large after a step in which $V$ was very close to the throat (where $DADX = 0$). A maximum of 50 iterative steps is allowed.
If the channel width is given (entry FINDXC), then $V$ is initialized to 100 cm, a value typical of axial positions where test panels might be installed in a channel. For each trial value of $V$, the profile half width $AR = Z(MBL)$ and slope $DADX = DZDX(MBL)$ are computed by calling the entry point GMAR3 of GEØMAR. The same Newton-Raphson formulas and convergence test are used as in the area ratio case.

3.22 Subroutine FRØZEQ

FRØZEQ is the routine which controls the computation of the frozen and equilibrium flow solutions. The frozen flow solution is generated when the routine is called through entry FRØZEN. Entry EQUIL(IPASS) gives the equilibrium solution.

3.22 Nondimensionalization of the Flow Variables

Throughout subroutine FRØZEQ, the calculations are carried out with most of the flow variables in nondimensional form. The relation between the Fortran symbols and the dimensional flow variables is summarized in the following formulas:

- $AFNX$: $A_e$, effective area ratio
- $CH$: $\frac{W_0h}{R_0T_0}$, nondimensional specific enthalpy
- $CHA$: $\frac{W_0h_0}{R_0T_0}$, nondimensional enthalpy in the reservoir
- $CT$: $T/T_0$, nondimensional temperature
- $CX$: $x$, axial nozzle coordinate (cm)
- $FLUX$: $\frac{\rho u}{\rho_0u_s}$, nondimensional mass flux
PRES \( \frac{P}{P_0} \), nondimensional pressure

\( \frac{\rho}{\rho_0} \), nondimensional density

SU \( \frac{u}{u_s} \), nondimensional velocity

In these definitions, the subscript 0 denotes conditions in the reservoir, \( R_0 \) is the universal gas constant, and \( u_s \) is a velocity defined by

\[
u_s = \sqrt{\frac{R_0 T_0}{W_0}}
\]

where \( W_0 \) is the molecular weight in the reservoir and \( R_0 \) is the universal gas constant in mechanical cgs units (erg/mole °K).

3.22.2 Determination of Frozen Sonic Conditions

When entry FRZEQ is called, subroutine FRZEQ determines the sonic conditions in the frozen flow before starting the main frozen flow solution. This calculation is done by the statements from entry FRZEQ down to entry EQUIL. A flowchart of this part of the subroutine is shown in figure 9. The criterion for locating the sonic point is that the mass flux \( m \) be a maximum. This condition follows from the continuity equation (243) and the fact that the flow area \( A \) is a minimum at the sonic throat. The temperature at the sonic point is determined as follows: the temperature is initialized to its reservoir value, and is then decremented repeatedly by the amount \( \Delta T = \frac{\Delta T}{0.01} \), where \( \Delta T \) is normally 0.01. At each of these temperatures, the corresponding mass flux in frozen flow is calculated by calling subroutines THERM and PRF. This procedure is continued until the mass flux passes its maximum. Then the temperature step is reduced and an iteration is carried out to locate the sonic-point temperature with high accuracy.

*The calculation of sonic conditions for the equilibrium solution is done in a separate subroutine, NRMAX.*
Figure 9. Flow Chart for Determination of Sonic Conditions
The algorithm for determining the sonic-point temperature (figure 9) is based upon consideration of a sequence of three equally-spaced temperatures T1 > T2 > T3 and the corresponding mass flux values F1, F2, F3. T1 and T2 are always chosen in such a way that F2 > F1; thus, T1 is definitely known to be higher than the sonic temperature. At each stage of the iteration, the rules for selecting the next triplet of temperatures T1', T2', T3' depend upon the value of F3 - F2. There are three cases, which are illustrated schematically in figure 10. In case A (F3 > F2), the three mass flux values form a monotonic sequence F3 > F2 > F1. In this case, the available data provide no indication that the point of maximum mass flux has been reached, although this may actually be the case, as shown by the dashed curve in the top section of the figure. In case A, T1' is set equal to T2, T2' to T3, and T3' is obtained by applying the same decrement T1' - T2' to T2'. In case B (F3 = F2), the sonic point is known to lie between T2 and T3. In this case, T1' is set equal to T2, the temperature decrement DELT = ΔT is cut in half, and the remaining two temperatures are set to T2' = T1' - ΔT and T3' = T2' - ΔT = T3. In case C (F3 < F2), the sonic point could lie between T1 and T2 or between T2 and T3. To ensure that T1' lies above the sonic temperature, it is set equal to T1. The decrement ΔT is halved as in case B, and T2' = T1' - ΔT, T3' = T2' - ΔT = T2. If the sonic point actually lies between T2 and T3, as illustrated by the solid curve in the bottom part of figure 10, the situation in the next step of the iteration will be as in case A.

At each step, the magnitude of DELT is compared with the criterion value TESTB = 10⁻⁵. If DELT < TESTB, the iteration has converged. A final improvement in the value of the sonic temperature is then obtained by fitting a parabola to the mass flux versus temperature data for the final triplet of points, and determining the maximum analytically based on this quadratic relation. The form assumed is

\[ F(T) = A + B(T-T_2) + C(T-T_2)^2 \]  \hspace{1cm} (77)
Case A: $F_3 > F_2$

Case B: $F_3 = F_2$

Case C: $F_3 < F_2$

Figure 10. Three Cases in the Algorithm for Finding the Throat Temperature
Substitution of the data points \( (T_1, F_1), (T_2, F_2), \) and \( (T_3, F_3) \) and solution for the coefficients \( A, B, C \) give
\[
F = F_2 - \frac{F_3 - F_1}{2(\Delta T)} (T - T_2) + \frac{F_1 + F_3 - 2F_2}{2(\Delta T)^2} (T - T_2)^2 \quad (78)
\]

Setting \( dF/dT = 0 \) then leads to the final result
\[
T_* = T_2 + \frac{(F_3 - F_1)(\Delta T)}{2(F_1 + F_3 - 2F_2)} \quad (79)
\]
If the quantity \( Z/F_2 = (F_1 + F_3 - 2F_2)/F_2 \) is less than or equal to \( 10^{-5} \), equation (77) is not used. In this case, \( T_* \) is simply set equal to \( T_2 \), to avoid possible errors due to loss of accuracy on the subtraction in \( Z \) (or a divide check in (79) if \( Z = 0 \)).

3.22.3 Main Flow Solution

The basic operation performed by subroutine \( \text{FRZEQ} \) is the calculation of the frozen and equilibrium flow solutions. In its essentials, this calculation is quite simple. The nondimensional temperature \( CT \) is decremented repeatedly, starting from its reservoir value \( (CT = 1.) \). At each value of the temperature, the other flow variables are computed, using subroutines \( \text{THERM} \) and \( \text{PRP} \) in the case of frozen flow and subroutine \( \text{NEWRAP} \) in the case of equilibrium flow. Then the position \( x \) in the nozzle at which the flow point occurs is determined from the known nozzle geometry and the value of the area ratio at the flow point, and the results are printed by calling entry \( \text{OUT2} \) of subroutine \( \text{OUT1} \).

The actual complexity of \( \text{FRZEQ} \), as illustrated by its flowchart (figure 11) is due to provisions for treating various options and regions of the solution. The control variables pertaining to some of these cases and conditions will be defined before the subroutine is discussed in detail:
Figure 11a. Flowchart of Subroutine FROZEQ (Part a)
Figure 11b. Flowchart of Subroutine FRSEQ (Part b)

```
220  Compute flow conditions at CT
     SKIP?
     ISW3B=0?
     ISWIC

310  CALL FINDX(AFNX,FLAG,CX)
     ISW3B=0?
     ISWIC

180  IF CX < XLMIG
     CALL SLVMAX

340  CALL AGSMIN(AFEC,DELAL,FLAG,AR,CX)
     ISW3B=0?
     ISWIC

180  IF IPASS=1 and ISWIC
     CALL SLVMAX

180  IF ISWIC=0 or AR > 4.*AFIC
     IF INSTRT>753.71
     ERROR SUPP
     RETURN

110  Restart solution
```
Figure 11b. Flowchart of subroutine FRJZ (Part d)
ISW3B = 0  Boundary layer is omitted.
ISW3B ≠ 0  Boundary layer is included.

IPASS = 1  Preliminary equilibrium solution from the reservoir to the sonic point, to determine the boundary layer displacement thickness at the throat for use in reservoir condition calculations.

IPASS = 2  Final flow solution.

ISØNIC = -1  Region upstream of the throat.
ISØNIC = 0  Sonic point.
ISØNIC = 1  Region downstream of the throat.

MØDLPT = .FALSE.  Regular point of the flow solution.
MØDLPT = .TRUE.  Iteration to find the conditions at a specified model point is in progress.

SKIP = .FALSE.  Regular flow point or model point.
SKIP = .TRUE.  Regular flow point just after a model point.

LASTPT = .FALSE.  Regular flow point or model point.
LASTPT = .TRUE.  Last point of solution.

The simplest case is that of a solution without the boundary layer and without any model points. In this case, FRØZEQ generates a point of the solution by traversing the loop outlined by the statement numbers 180, 190, 220, 310, 410, 420, 440, 450, 550, 580, 600. The temperature decrement DELTV is equal to the input value of DELT1 (normally 0.01) in this case, until the difference X∅ - XP in spatial position between two successive flow points exceeds 10 cm. Once this has happened, the unnumbered statements between 180 and 190 reduce DELTV so as to maintain a spatial step size of approximately 10 cm. This variation of DELTV occurs far downstream of the throat, in the region where the free stream temperature has fallen to a small fraction of the reservoir
temperature. DELTV is not allowed to fall below 0.02 DELT1. These adjustments of DELTV in the region far downstream are also made when the boundary layer is included in the solution.

The second statement below 190 tests whether the temperature CT is below the sonic temperature CTMAX, i.e., whether the solution has passed the throat. If it has, in the case without the boundary layer, ISNONIC is reset from -1 to +1, and the indicator FLAG for upstream and downstream solutions in the geometry routines is reset from -1 to +1. Once the flow conditions for the temperature CT have been computed, the spatial position in the nozzle is determined by calling subroutine FINDX(AFNX, FLAG, CX) (statement 310). This subroutine determines the upstream (FLAG = -1) or downstream (FLAG = +1) position CX at which the geometric area ratio is equal to AFNX.

If the boundary layer is included, the input value of DELT1 is saved in the location DELTSV, and DELT1 is reset to 0.049(1. - CTMAX) during initialization. This ensures that there will be 20 flow points upstream of the throat. In some cases with CTMAX ~ 0.9, the normal input value of DELT1 = 0.01 would give only about 10 points above the throat, which would adversely affect the accuracy of the numerical integration in the boundary layer routine BLAYER. Once the throat has been reached, DELT1 is restored to its original value by the statements below 320.

When the boundary layer is included, the effective area ratio $A_e = AFNX$ is no longer equal to the geometric area ratio $A_g = AR$, because of the boundary layer displacement thickness, $\delta^*$. The relation between these two area ratios (Section 4.4 of Volume I) involves the displacement thickness at the throat, $\delta^*$. Since this is not known initially, the spatial position CX corresponding to each flow point is calculated on the assumption that $A_g \approx A_e$ until the throat has been reached. These calculations are done using subroutine FINDX, as in the case without the boundary layer. A special flow point is computed at the throat (ISNONIC = 0, CT = CTSTAR, CX = 0.) in order to determine the quantity
1 - \( \delta^* \) = \( \phi_{MDST} \) (statement 390) for the nozzle (or two such values in the case of a channel). Then, beginning at the third point beyond the throat, \( CX \) is computed taking the displacement thickness into account, using subroutine \( AGS\_LN \). In statement 340, \( AGS\_LN \) is called with the effective area ratio \( AFNX \) for the current step and the displacement thickness \( DELBL \) for the preceding step. The resulting \( CX \) value is used in the boundary layer calculations (statement below 370). Subroutine \( AGS\_LN \) is then called again, with the same \( AFNX \) and the new \( DELBL \) produced by the call to \( BLAYER \). This second call to \( AGS\_LN \) gives an improved value of \( CX \).

In some problems, at large distances downstream of the throat, the displacement thickness becomes large enough to have a major effect on the effective area ratio. Under such circumstances, the coupling between the boundary layer and the inviscid flow solution can lead to computational instability. The boundary layer routine \( BLAYER \) contains a smoothing algorithm intended to suppress such instability (Section 5.11 of Volume I). This algorithm involves a parameter \( W \), preset to a value \( WSAVE \) which usually prevents instability. However, occasionally instability develops in spite of the smoothing. In the frozen and equilibrium solutions, the symptom of such instability is that \( CX \) begins to decrease with decreasing temperature. When this occurs, subroutine \( FR\_ZEQ \) attempts to produce a valid solution by cutting the stabilizing parameter \( W \) in half and restarting the solution in the reservoir. This operation is accomplished by coding between statements 340 and 350. If instability has been encountered after three successive restarts, the solution is abandoned and error dumps are written. Also, restarting is not attempted if the geometric area ratio is more than 4 times greater than the effective area ratio at the point where \( CX \) decreases, as in this case the code is probably predicting actual choking of the flow by boundary layer growth.

3.22.4 Model Points

Model points are input-specified spatial locations, in the downstream region of the nozzle, at which special calculations are to be performed. Since model points are
specified by values of $CX$, while points of the equilibrium or frozen flow solution are specified by values of the temperature, an iteration is required to find the temperature $CT$ corresponding to each model point.

During the initialization, and after the completion of each model point calculation, subroutine NEXTMP is called to determine the location $XMODEL$ of the next model point. During the main flow solution, after $CX$ has been determined for each flow point, statement 450 tests whether $CX$ is greater than or equal to $XMODEL$. If not, the flow solution proceeds normally. If $CX \geq XMODEL$, the iteration to determine the CT value corresponding to $CX = XMODEL$ is started. While this iteration is in progress, the indicator $MODELPT$ is .TRUE..

To avoid complicating the programming of subroutine BLAYER, and to save computer time, BLAYER is not called during the model point iteration. Instead, the boundary layer properties at the model point are approximated by linear interpolation of the data at the preceding and current main flow points, and at each cycle of the iteration subroutine TRANSP is called (statement 360) to determine the transport properties at the current temperature.

The iterative algorithm for improving the estimate of $CT$ corresponding to $XMODEL$ is a numerical Newton-Raphson technique embodied in statement 490:

$$ T = T_1 + \left( T_2 - T_1 \right) \frac{X_m - X_1}{X_2 - X_1} \quad (80) $$

where $T_2$ is the temperature for which calculations were done in the current step, $T_1$ is the temperature in the preceding iterative step, $X_2$ and $X_1$ are the $CX$ values corresponding to $T_2$ and $T_1$, and $X_m$ is $XMODEL$.

The convergence criterion for the iteration is that $CX$ differ from $XMODEL$ by no more than 0.0025 cm or 0.001 inch. A maximum of 10 iterations is allowed. If convergence has not been achieved after 10 iterations, a diagnostic message
is written and the data from the final iteration are used. When the iteration has thus been terminated by either convergence of nonconvergence, the results for the flow conditions at the model point are printed by calling OUT2, MÖDEL is reset to .FALSE., and subroutine MÖDEL is called to perform the required calculations of conditions on models. Also, the indicator SKIP is set to .TRUE., and the next XMÖDEL is determined by calling NEXTMP. Before resuming the main flow solution, FRÉEQ checks whether the next XMÖDEL is less than or equal to the CX for the current main solution point. If so, the model point iteration is started again to find the flow conditions at the new XMÖDEL.

Once all of the XMÖDEL values less than or equal to the CX for the current main flow point have been dealt with, the flow conditions for the current flow point are recomputed and printed. The .TRUE. value of the indicator SKIP provides for this recomputation by causing a transfer to statement 220 instead of 550 just below statement 450. Also for SKIP = .TRUE., the determination of CX and the call to BLAYER are bypassed below 220. SKIP is then reset to .FALSE., the results are printed by calling OUT2, and the main flow solution is resumed.

When a CX greater than or equal to CXMAX is reached, XMÖDEL is set equal to CXMAX and the indicator LASTPT is set to .TRUE.. The model point iteration then provides the flow conditions at the position CXMAX. After the iteration has converged and the results have been printed, for LASTPT =.TRUE., control is transferred to statement 610. For nozzle flow problems (NPRFLS = 1), the RETURN is then executed. For channel problems (NPRFLS = 2), subroutine MÖDEL is first called with AXISYM = .TRUE. to provide calculation of the stagnation conditions in the channel exit plane. Stagnation condition calculations are suppressed for channels, with this one exception, because stagnation point models and Pitot probes are not used inside channels.

3.23 Subroutine GE®M

The function of GE®M is to calculate the density, the effective area ratio and the derivative of one or the other of these quantities at each position x along the nozzle during
the nonequilibrium flow solution. Two different methods of
calculation are used in different parts of the flow solution.
During the solution by the perturbation method, and during
the direct integration of the rate equations in the region
beyond the call to THRΩAT, the calculation of the effective area
ratio is based directly on the nozzle geometry and the boundary
layer displacement thickness, if any, and the density is then
calculated from the continuity equation. During the numerical
integration by the inverse method upstream of the call to THRΩAT,
the density is obtained from the analytical density-area relation
I(383):
\[
\left( \frac{\rho}{\rho_0} \right)^2 A_e \left[ 1 - \left( \frac{\rho}{\rho_0} \right)^\alpha \right] = C
\]  
(81)
and the effective area ratio is then obtained from the con-
tinuity equation I(266). The constants \(\alpha\) and \(C\) in (81)
are determined in the main program from sonic-point condi-
tions, prior to the first call to GEΩM, as explained in Sec-
tion 7.4 of Volume I and in Section 3.1 of the present re-
port.

The overall logical structure of GEΩM is diagrammed in
figure 12. Immediately upon entry, the subroutine calls
GEΩMAR to compute the geometric area ratio \(S_1\) and its deri-
ivative \(S_2\) at the current position \(C_X\). If the nonequilibrium
flow solution being computed includes the boundary layer
(ISW3B \(\neq 0\)), then AESΩLN is called to convert \(S_1\) into the
effective area ratio and \(S_2\) into the derivative of the ef-
fective area ratio, based on the current displacement thick-
ness.

If the flow solution is beyond the call to THRΩAT
(IUPD = 0) or if the perturbation technique is in use
(INEQ = 0), the effective area ratio AFNX is set equal to
\(S_1\) and the nondimensional density RHΩ is calculated from
AFNX using the continuity equation. Also, the derivative
DLΩGA = \(d / n \ A_e / dx\) is computed from \(S_1\) and \(S_2\).

On the other hand, if the numerical integration tech-
nique and the inverse method are being used to generate the
flow solution (INEQ = 1 and IUPD = 1), then the nondimensional
Figure 12. Flowchart of Subroutine GE@M

-93-
density \( \rho \) is calculated from S1 by solving the density-area ratio relation (81), and the effective area ratio \( A_{FX} \) is recomputed from \( \rho \) using the continuity equation. The derivative \( DL@X = \frac{d \ln (\rho/\rho_0)}{dx} \) is then computed for use in the rate equations (see the discussion of subroutine EXACT above).

The calculation of \( \rho \) from equation (81) is based on the following analysis. Equation (81) can be solved for \( A_e \):

\[
A_e = \frac{\rho_0}{\rho} \sqrt{\frac{C}{1 - (\rho/\rho_0)^\alpha}}
\]  
(82)

The form of this relation is illustrated in figure 13, for the case \( \alpha = 0.2, C = 0.035049 \). The corresponding density ratio at the sonic point is \( \rho^*/\rho_0 = 0.62092 \). The basic computational problem in GE\( \Phi \)M is to solve (81) or (82) for \( \rho/\rho_0 \), based on a given value of \( A_e \). Except near the throat, the solution is carried out using the Newton-Raphson method, i.e., from a given estimate \( \rho_p \) of \( \rho \), an improved estimate \( \rho_n \) is obtained as follows:

\[
A_p = \frac{\rho_0}{\rho_p} \sqrt{\frac{C}{1 - (\rho_p/\rho_0)^\alpha}}
\]  
(83a)

\[
\rho_0 \left( \frac{dA}{d\rho} \right)_p = \frac{\rho_0 A_p}{\rho_p} \cdot \frac{(1 + \frac{1}{2} \alpha)(\rho_p/\rho_0)^\alpha - 1}{1 - (\rho_p/\rho_0)^\alpha}
\]  
(83b)

\[
\rho_n = \rho_p + \frac{A_e - A_p}{(dA/d\rho)_p}
\]  
(83c)

Here \( A_p \) is the \( A_e \) value calculated from (82) for \( \rho = \rho_p \); \( (dA/d\rho)_p \) is the derivative of (82) evaluated at \( \rho = \rho_p \)
and equation (83c) is the Newton-Raphson formula.
Figure 13. Approximations to the Density-Area Ratio Relation
Examination of equations (83) reveals several potential problems. At the sonic point \((A_e = 1)\), the derivative \(dA_e/d\rho\) is zero, so that the second term in (83c) is undefined. Near the throat, \((dA_e/d\rho)\) is very small. If a trial value \(\rho_p\) is near the throat, and nearer the throat than the solution of (82), the Newton-Raphson formula (83c) can give an "improved" \(\rho_n\) which is much less accurate than \(\rho_p\) and which may even be outside the permissible range \(0 \leq \rho \leq 1\). To deal with these problems reliably, GEQK uses the following algorithm:

1. If \(A_e \leq 1\), \(\rho = \rho^*\) (throat point)

2. If \(1 < A_e \leq 1.001\) (throat region), \(\rho\) is calculated from a Taylor series expansion of (82) around the throat point:

\[
A_e = 1 + \frac{1}{2} \left( \frac{d^2 A_e}{d\rho^2} \right) \rho^* (\rho - \rho^*)^2 + \frac{1}{6} \left( \frac{d^3 A_e}{d\rho^3} \right) \rho^* (\rho - \rho^*)^3
= 1 + \frac{\alpha}{2c} \left( \frac{\rho}{\rho_0} - \frac{\rho^*}{\rho_0} \right)^2 + \frac{\alpha + 1}{6c\sqrt{\alpha+2}} \frac{\alpha^{3/2}}{\rho_0} \left( \frac{\rho}{\rho_0} - \frac{\rho^*}{\rho_0} \right)^3
\]

The coefficients in this equation have been evaluated by differentiation of (82) and use of the sonic-point conditions \(\rho = \rho^*\), \(A_e = 1\), \((dA_e/d\rho)_* = 0\). In the throat region, where (84) is used, the cubic term is small. If this term were neglected, the solution of (84) would be

\[
\frac{\rho}{\rho_0} - \frac{\rho^*}{\rho_0} = S \frac{\sqrt{2c}}{\alpha} (A_e - 1) = S \cdot (\delta \rho)
\]

where \(S\) is +1 upstream and -1 downstream of the throat. When the cubic term is taken into account, the solution of (84) can be approximated by
\[
\frac{\rho}{\rho_0} \approx \frac{\rho^*}{\rho_0} = \frac{2C(A_e - 1)}{\alpha} - \frac{5(\alpha + 1)}{3} \sqrt{\frac{\alpha}{C(\alpha + 2)}} \cdot (\delta \rho)^{\frac{3}{2}}
\]

where \( \delta \rho \) is defined by the second equality in (85).

Equation (86) is used to calculate \( \rho / \rho_0 \) when \( A_e < 1.001 \).

(3) If \( A_e > 1.001 \) (general case), the Newton-Raphson procedure (83) is used. To ensure its stability, pains are taken to select a first estimate of \( \rho \) which is farther from \( \rho^* \) than the solution point. Since the relation (82) is everywhere concave upward (figure 13), the Newton-Raphson formula undercorrects \( \rho \) whenever the estimate \( \rho_0 \) lies too far from \( \rho^* \). Thus, if the first estimate \( \rho_0 \) is too far from \( \rho^* \) the iteration converges to the solution without ever overshooting it. Under these circumstances, no instability can develop. To obtain such an initial estimate, the subroutine uses three approximations to (82), all of which lie below (82). One of these approximations is (85). The second is an approximation for \( \rho / \rho_0 \sim 1 \), namely

\[
A_e \sim \sqrt{\frac{C}{1 - (\rho / \rho_0)^{\alpha}}}
\]

from which

\[
\frac{\rho}{\rho_0} \sim (1 - \frac{C}{A_e^2})^{1/\alpha}
\]

(87)

The third is an approximation for \( \rho \sim 0 \),

\[
A_e \sim \sqrt{C / (\rho / \rho_0)}
\]

from which

\[
\frac{\rho}{\rho_0} \sim \sqrt{C / A_e}
\]

(88)

In the upstream region \( (dA_e / dx < 0) \), the initial estimate is chosen to be

\[
\frac{\rho_0}{\rho^*} = \min \left[ \frac{\rho^*}{\rho_0} + \frac{\sqrt{2C(A_e - 1)}}{\alpha}, \left(1 - \frac{C}{A_e^2}\right)^{1/\alpha} \right]
\]

(89)
In the downstream region \((dA_e/ dx > 0)\), it is

\[
\frac{\rho p}{\rho_0} = \max \left[ \frac{\rho}{\rho_0} - \frac{\sqrt{2C(A_e - 1)}}{\alpha}, \frac{\sqrt{C}}{A_e} \right]
\]

(90)

Once the solution for \(\rho\) has been obtained, the derivative \(d\ln \rho/ dx\) is calculated. Normally this is determined from the formula

\[
\frac{d\ln \rho}{dx} = \frac{2C}{\alpha (\frac{\rho A_e}{\rho_0})^2 - (\alpha + 2)C} \cdot \frac{d\ln A_e}{dx}
\]

(91)

which can be obtained by differentiating (82). The factor \(d\ln A_e/ dx\) on the right is evaluated as \(S2/ S1\), based on the specified nozzle geometry and displacement thickness. The throat point \((A_e = 1)\) is a special case, because there the denominator of (91) is zero and \(dA_e/ dx = 0\), so that (91) is of the form \(0/0\). At the throat, \(d\ln \rho/ dx\) must be evaluated from (91) using L'Hopital's rule:

\[
\begin{align*}
\frac{d\ln \rho}{dx}_{A_e = 1} &= \frac{2C}{\alpha} \lim_{A_e \to 1} \frac{2\alpha}{\rho_0} \left( \frac{dA_e}{dx} + A_e \frac{d\rho}{dx} \right) \\
\frac{d\ln A_e}{dx} &= \frac{d^2 \ln A_e}{dx^2}
\end{align*}
\]

Since \((dA_e/ dx)_* = 0\), this gives

\[
\frac{d\ln \rho}{dx}_* = -\sqrt{\frac{C (d^2 \ln A_e/ dx^2)_*}{\alpha}} \frac{\rho_*}{\rho_0}
\]

(92)

The area derivative on the right hand side has to be evaluated from the specified nozzle geometry. In general,

\[
\frac{d^2 \ln A_e}{dx^2}_* = \frac{d^2 A_e}{dx^2}_*
\]

(93)
by virtue of the conditions \( A_{e*} = 1, \ (dA_e/dx)_* = 0. \) To proceed further, it is necessary to consider the three geometry options separately:

(1) Two-dimensional nozzle

In this case, from equation (126) with \( y_* \) replaced by \( y_0 \) as usual,

\[
\left( \frac{d^2A_e}{dx^2} \right)_* = \frac{1}{y_0} \left( \frac{d^2y}{dx^2} \right)_* - \frac{1}{y_0} \left( \frac{d^2\delta^*}{dx^2} \right)_* \]

The term \( (d^2 \delta^*/dx^2)_* \) is small, because the displacement thickness \( \delta^* \) is small and slowly varying at the throat. This term is therefore neglected. Since the throat sections of the profile fit are always of the form (122) (shape 2, circular arc convex toward the axis), the remaining term in the numerator of (94) is given by

\[
1 \left( \frac{d^2y}{dx^2} \right)_* = \frac{1}{y_0} \left( \frac{d^2y}{dx^2} \right)_0 = \frac{1}{y_0^2} \]

The second equality is obtained by noting that \( P_2=0 \) for a throat section. Thus,

\[
\left( \frac{d^2A_e}{dx^2} \right)_* = \frac{1}{y_0^2(1 - \delta^*/y_0)} \]

(2) Axisymmetric nozzle

From equation (131) since \( (dA_e/dx)_* = 0 \) and \( A_{e*} = 1, \)

\[
\left( \frac{d^2A_e}{dx^2} \right)_* = \frac{1}{1 - \delta^*/y_0} \left[ \frac{1}{\sqrt{A_{g*}}} \left( \frac{d^2A_g}{dx^2} \right)_* \right. \]

\[
- \frac{1}{2A_{g*}^{3/2}} \left( \frac{dA_g}{dx} \right)_* \frac{2}{y_0} \left( \frac{d^2\delta^*}{dx^2} \right)_* \]

(96)
Now $dA_g/dx$ is very small at the sonic point (which is quite near the throat), so that the square of this quantity can be neglected. The quantity $(d^2 \delta^*/dx^2)_*$ is also neglected for reasons discussed above. Hence,

$$\frac{d^2 A_e}{dx^2} \simeq \frac{(d^2 A_g/dx^2)_*}{\sqrt{A_g}(1 - \delta^*/y_0)}.$$  \hfill (97)

Evaluation of $(d^2 A_g/dx^2)_*$ using $I(116)$ and $I(122)$ then gives

$$\frac{d^2 A_e}{dx^2} \simeq \frac{2}{y_0^3(1 - \delta^*/y_0) \sqrt{A_g}}.$$  \hfill (98)

(3) Rectangular channel

From equation $I(134)$,

$$\frac{d^2 A_e}{dx^2} = \frac{1}{D} \left[ (y - \delta^*(1) \frac{d^2 z}{dx^2} - \frac{d^2 \delta^*}{dx^2} + (z - \delta^*(2) \frac{d^2 y}{dx^2} - \frac{d^2 \delta^*}{dx^2}) 
+ 2 \left( \frac{dy}{dx} - \frac{d \delta^*}{dx} \right) \left( \frac{dz}{dx} - \frac{d \delta^*}{dx} \right) \right]^*$$  \hfill (99)

where

$$D = (y_0 - \delta^*_1)(z_0 - \delta^*_2).$$  \hfill (100)

At the sonic point, the quantities $\left( \frac{dy}{dx} - \frac{d \delta^*}{dx} \right)$ and $\left( \frac{dz}{dx} - \frac{d \delta^*}{dx} \right)$ are small, so that their product is considered negligible. The quantities $(d^2 \delta^*_1/dx^2)_*$ are also neglected, as in the other options. Also, $y^*_* \simeq y_0$ and $z^*_* \simeq z_0$. Hence,
(\frac{d^2A_e}{dx^2}) = \frac{(y_0 - \delta_{1*})(d^2z/dx^2)_* + (z_0 - \delta_{2*})(d^2y/dx^2)_*}{(y_0 - \delta_{1*})(z_0 - \delta_{2*})}

(101)

Evaluation of \((d^2z/dx^2)_*\) and \((d^2y/dx^2)_*\) from \(I(122)\) gives, finally,

\(\frac{d^2A}{dx^2} = \frac{1}{(z_0 - \delta_{2*})P_{3z}} + \frac{1}{(y_0 - \delta_{1*})P_{3y}}\)

(102)

where \(P_{3z}\) is the \(P_3\) parameter for the \(z\) profile and \(P_{3y}\) that for the \(y\) profile.

### 3.24 Subroutine GEOMAR

GEOMAR is the basic geometry routine in the NATA code. Through its various entries, it provides calculations of the geometric area ratio and its derivative and of profile ordinates and their derivatives both for nozzles and for rectangular channels.

The subroutine has 4 entries, with the following calls:

- \(\text{GEOMAR}(X, \text{ARATI} \delta, \text{DERIVA})\) \hspace{1cm} (\text{IENTRY} = 1)
- \(\text{GMAR}(X, Y)\) \hspace{1cm} (\text{IENTRY} = 2)
- \(\text{GMAR2}(X, Y, Z)\) \hspace{1cm} (\text{IENTRY} = 3)
- \(\text{GMAR3}(X, \text{DYDX}, \text{DZDX}, Y, Z)\) \hspace{1cm} (\text{IENTRY} = 4)

The arguments have the following meanings:

- \(X\) = axial coordinate in nozzle (cm)
- \(\text{ARATI} \delta\) = geometric area ratio
- \(\text{DERIVA}\) = derivative of geometric area ratio (cm\(^{-1}\))
- \(Y\) = profile ordinate (cm)
For each profile (I) the subroutine first determines the profile section (J) in which the specified value of X lies. This is the first section for which X < ATPI(J, I). The routine then calculates the profile ordinate from equation I(121), I(122), or I(123) depending on ISHAPE(J, I), and it required also calculates the derivative of the profile ordinate using the derivatives of these formulas:

(1) Straight Line, ISHAPE(J, I) = 1

\[
\frac{dy}{dx} = \frac{P_2}{P_2} \quad (103)
\]

(2) Circular Arc Convex Downward, ISHAPE(J, I) = 2

\[
\frac{dy}{dx} = \frac{x - P_2}{\sqrt{P_3^2 - (x - P_2)^2}} \quad (104)
\]

(3) Circular Arc Concave Downward, ISHAPE(J, I) = 3

\[
\frac{dy}{dx} = -\frac{x - P_2}{\sqrt{P_3^2 - (x - P_2)^2}} \quad (105)
\]

The subroutine then calculates the geometric area ratio, if required, from equation I(118) for a two-dimensional nozzle (NPRFLS = 1, JDIM = 0), I(116) for an axisymmetric nozzle (NPRFLS = 1, JDIM = 1), or I(120) for a rectangular channel (NPRFLS = 2). It calculates the derivative of the area ratio from the derivatives of these formulas, i.e.,

\[
\frac{\delta A_y}{dx} = \frac{1}{y_0} \frac{dy}{dx} \quad (2D \text{ nozzle}) \quad (106)
\]
\[
\frac{dA_q}{dx} = \frac{2y}{y_0^2} \frac{dy}{dx} \quad \text{(axisymmetric nozzle)} \quad (107)
\]
\[
\frac{dA_g}{dx} = \frac{y \, dz + z \, dy}{y_0^2} \quad \text{(channel)} \quad (108)
\]

Before executing the RETURN, the routine rescales all of the computed results using the rescaling factor RSA. The area ratio and its derivative are multiplied by RSA in the case of a 2D nozzle, but in an axisymmetric nozzle or channel they are multiplied by \(\sqrt{\text{RSA}}\). The factor RSA is initialized to 1.0 and is never changed in equilibrium and frozen flow solutions. In nonequilibrium flow solutions, RSA is reset in subroutine THR\(\theta\)AT so as to make the effective area ratio continuous between the upstream solution by the inverse method and the downstream direct solution.

### 3.25 Subroutine INGAS

This subroutine has two entry points, INGAS and INTA. The first part of the routine, which is entered by calling INGAS, sets up several arrays which are used in the chemical description of the gas mixture, as discussed in Section 2.1 of Volume I. The mathematical and Fortran notations for these arrays correspond as follows:

- \(A_{ki}\) \(\Rightarrow\) AIN(K,I)
- \(\bar{\nu}_{i-c,j}\) \(\Rightarrow\) CDIJ(I,J) \(\quad (I = i - c)\)
- \(-1 + \nu_{i-c}^*\) \(\Rightarrow\) BET(I) \(\quad (I = i - c)\)
- \(q_i\) \(\Rightarrow\) QM(I)
- \(W_i\) \(\Rightarrow\) CGI(I)

In the subroutine, the array AIN is first initialized to the square submatrix \(\overline{\alpha}_{ii}\) of the matrix \(\alpha_{ij}\) with \(i = 1\) to \(c\), where \(\alpha_{ij}\) is the number of atoms of the \(j\)th element...
per molecule of the \( i \)th species, and \( c \) denotes the number of elements in the system. The inverse matrix \( A_{ki} \) of \( \alpha_{ij} \) is then computed and stored in AIN by calling subroutine MATINV.

Next, the matrix \( \bar{\nu} \), which gives the composition of the dependent species in terms of the components, is then computed by evaluating equation (6), and is stored in CDIJ. Then the singly subscripted array \( -l + \nu_{i-c}^* \), where \( \nu^* \) is defined by equation (15), is computed in BET. The array \( q_i \) giving the invariant composition of the gas in terms of the independent species is calculated using equations (9) and (10); it is denoted by QM(I). Finally, the species molecular weights \( W_i = CGI(I) \) are computed as follows:

\[
\begin{align*}
C_i &= \sum_{j=1}^{c} \alpha_{ij} A_j \\
W_i &= C_i \quad (i = 1, \ldots, c) \\
W_i &= \sum_{k=1}^{c} \bar{\nu}_{i-c,k} W_k \quad (i = c + 1, \ldots, n)
\end{align*}
\]

For the rationale of equation (110b), see equation (5) and its associated discussion. The molecular weights of the dependent species \( i = c + 1, \ldots, n \) could also be calculated directly using equation (109), in place of (110b).

The purpose of the calculations following entry INTA is to compute the equilibrium conditions in the upstream reservoir. The actual thermochemical equilibrium calculation is performed by subroutine EQCALC(CTAP,PRESA). The argument CTAP is the reservoir temperature in degrees Kelvin, and PRESA is the reservoir pressure in atmospheres.
The results of the calculations are communicated from EQCALC to INGAS through the common blocks /EQC/ and /EQC2/.

3.26 Subroutine INIT

Subroutine INIT initializes a number of control parameters and nondimensionalizes the species thermochemical data using the reservoir temperature. Since the coding is straightforward, it will not be discussed.

3.27 Subroutine KANDMU

Subroutine KANDMU calculates quantities proportional to the viscosity and the translational thermal conductivity of the gas mixture from equations I(85) to I(87). The first section of the subroutine, down through statement 20, computes the quantities $(25/8)\sqrt{T} A_i^{(K)}$ and $(25/8)\sqrt{T} A_i^{(\kappa)}$ defined by equation I(87), for all species $i$ in the mixture. The value of $(25/8)\sqrt{T} A_i^{(\kappa)}$ is obtained simply by dividing the factor $Z M 2(I)$ out of the quantity $B(I,2)$ computed previously in subroutine KINT, equation (113) below, and is stored again in the array $B(I,2)$, while the quantity $(25/8)\sqrt{T} A_i^{(K)}$ is computed directly from I(87) using the matrix elements $Q(3,I,J)$ calculated previously in subroutine TRANSP. The value of $(25/8)\sqrt{T} A_i^{(K)}$ is stored in the location $B(I,1)$, replacing the previous quantity set in KINT, which is no longer needed. The reciprocals of the elements $B(I,L)$ are also computed and stored in $BR(I,L)$.

The next section of subroutine KANDMU, from statement 20 down through statement 70, computes the quantities

$$A(1) = (25/8)\sqrt{T} \pi^{(K)}$$

and

$$A(2) = (25/8)\sqrt{T} \bar{\alpha}^{(\kappa)}$$

from equation I(86), using the values of the matrix elements $a_{ij}^{(K)}$ and $a_{ij}^{(\kappa)}$ which were previously stored in the lower halves ($I > J$) of the matrices $Q(1,I,J)$ and $Q(2,I,J)$.
by subroutine TRANSP. Since the values of $a^{(\alpha)}$ for the viscosity and thermal conductivity are both computed from the same equation, the same coding can be used here for both properties, with the index $L = 1$ indicating values computed for the thermal conductivity and $L = 2$ values computed for the viscosity. In connection with this computation, it may be noted that the sum in equation I(86) need only be carried over values of $I > J$, since the matrix $a_{ij}^{(\alpha)}$ is symmetric and the diagonal terms do not contribute to the sum.

The final loop in subroutine KANDMU down through statement 90 computes the values of the subroutine arguments $ZK(1)$ and $ZK(2)$, which are proportional respectively to the translational thermal conductivity and the viscosity of the gas mixture. These values are computed from equation I(85), using the values of the elements $B(I,L)$ and $A(L)$ which were computed earlier in the subroutine.

3.28 Subroutine KINT

The statements down through statement 10 of subroutine KINT compute the quantities

$$B(I,1) \equiv \frac{1}{k} \sqrt{T} \sum_{j=1}^{n} x_j \Delta_{ij}^{(1)}$$

and

$$B(I,2) \equiv \frac{5}{6k} \sqrt{T} \sum_{j=1}^{n} x_j \Delta_{ij}^{(2)} = \frac{25}{8} \sqrt{T} ZM2(I) \sum_{j=1}^{n} x_j a_{ij}^{(\mu)}$$

which are required in the transport property calculations. Because of the way in which the data $\Delta_{ij}^{(k)}$ are stored in the Q array by subroutine TRANSP, it is necessary to treat the terms with $i < j$ and with $i > j$ separately in these
computations; thus the statement preceding 10 carries out
the sum over terms with \( i > j \) and statement 10 the sum
over terms with \( i < j \). The diagonal terms with \( i = j \) were
added previously during the initialization of the sum by
subroutine TRANSP.

Following the computation of the \( B(I,1) \) and \( B(I,2) \),
subroutine KINT uses the computed values of the \( B(I,1) \) in
the small loop ending at statement 30 to calculate the
value of the argument \( ZKINT = K_{int} / \sqrt{T} \) in units of milli-
watts/cm\(^{-2}\)-K\(^{3/2} \), where \( K_{int} \) is the internal component of the
thermal conductivity of the gas given by equation I(95).

3.29 Subroutine LIST

Subroutine LIST prints out certain portions of the
input data for each case so as to provide a description
of the problem which is to be solved by the code. Subrou-
tine READ also provides some output of similar character.

Portions of the output normally produced by LIST are
omitted if they would simply repeat the corresponding out-
put for a previous case. If \( SUPG = .TRUE. \), the gas model
is the same as in the preceding case, and the description
of the gas model is skipped. If \( SUPG = .FALSE. \), the fol-
lowing tables are printed:

(1) The elemental composition of the gas in terms
of "atom fractions", defined as the number of
gram-atoms of each element per mole of the
mixture of cold species. This table also lists
the atomic weights assumed for the elements.

(2) The data defining the forward rate constant
for each of the reactions. This table also
includes the criterion value CCHI for switch-
ing from the perturbation technique to num-
erical integration, and the third body matrix
for the reactions involving third bodies.
(3) A table which defines the chemical formula for each species by giving the $\alpha_{ij} = LPIJ(I,J)$ matrix. A separate column of this table contains 1 if a thermo fit is available for the species, 0 if not.

(4) The matrix $\gamma_{ij}'$ of the stoichiometric coefficients for the product side of all the reactions.

(5) The matrix $\gamma_{ij}'$ of the stoichiometric coefficients for the reactant side of all the reactions.

(6) A table listing the data for calculations of species thermal properties based on the physical model.

(7) A table listing the electronic energy levels and degeneracies for all the species for which physical model data are provided.

(8) A table of the thermo-fit coefficients for those species for which thermo fits are provided.

If an electronic nonequilibrium gas model is being used (NT=2), then a table giving the extra reaction data required by this model is inserted before the thermo-fit table (8).

After the gas model description has been printed out, LIST prints a summary of the input data for model and wedge calculations.

3.30 Subroutine MATINV

This is a matrix inversion subroutine. It is called by INGAS to compute the inverse $A_{ki}$ of the matrix $\gamma_{ij}$; see equation I(3) and associated discussion. MATINV computes the inverse of matrix $A$ using the Gauss-Jordan reduction with the maximum pivot strategy (ref. 8).

3.31 Subroutine M\O DEL

Subroutine M\O DEL is the principal routine for calculations of conditions on models in the NATA code. M\O DEL performs normal-shock calculations, computes conditions at the
inviscid stagnation point, calculates stagnation point heat fluxes, and provides output of these results. In addition, if conditions on wedge models are to be calculated, MODDEL calls subroutine WEDGE. The overall structure of MODDEL is diagrammed in figure 14.

3.31.1 General Description

Immediately upon entry, MODDEL performs some checks to determine whether model condition calculations should be done. Such calculations are skipped if both of the logical variables AXISYM and WEDGEM are false, or if the freestream Mach number is less than 1.5. The latter restriction is imposed because the normal shock solution fails to converge when the Mach number is below some value in the range from 1.5 to about 2. Models are not normally tested at such low Mach numbers in arc-heated wind tunnels because the corresponding nozzle diameter is too small to accommodate models of reasonable size.

If model calculations are to be done, subroutine ELTIME is called with the argument IP = 0 to record the execution time at the beginning of the model calculations. ELTIME is called again, just prior to the RETURN, to determine the CPU time elapsed during the model calculations. This time is printed out. It is usually about 1 to 3 seconds.

Before beginning the normal shock solution, MODDEL sets a number of flow quantities which depend upon the freestream flow conditions and are thus constant for a given call to MODDEL. When MODDEL is first called in a given flow solution, the equilibrium and frozen shock density ratios EPSILON and EPSF have the value zero, set in the calling routine (FRZEQ or NONEQ). These ratios are reset to the value 0.01. When MODDEL is called with EPSILON = 0, it computes the Prandtl number at the model surface temperature, PRW, based on the cold species mole fractions QPJ(K). This value of PRW thus assumes that the gas at the model surface is in a state of thermochemical equilibrium at the surface temperature.

Most of the calculations in MODDEL are contained within a large DØ loop with the DØ index ISØLN, beginning below
Figure 14. Overall Flowchart of Subroutine MDEL
statement 40 and ending at 440. For ISGLN = 1, the normal shock is calculated assuming chemical equilibrium behind the shock. For ISGLN = 2, the flow across the shock is assumed to be chemically frozen. If the control variable FSTAG is 0, the frozen shock calculations are skipped unless wedge calculations are to be done (WEDGEM = .TRUE.). If wedge calculations are called for, the frozen shock solution is carried out, regardless of the value of FSTAG, to provide the frozen stagnation temperature which is required for use in subroutine WEDGE. If FSTAG is negative, the equilibrium shock calculations are skipped.

For either value of ISGLN, the conditions behind the normal shock and those at the inviscid stagnation point are calculated using methods discussed below in Sections 3.3.1.2 and 3.3.1.3. Then the stagnation-point velocity gradient parameter \( \left( \frac{R_p}{u_1} \right) \left( \frac{d u_e}{d x} \right)_e \), denoted by VGP(L), is computed for spherical or cylindrical nosed models (L = 1) and for flat-faced models (L = 2). The ratio SOD(L) of the shock standoff distance to the model radius is also calculated for L = 1 and 2. These calculations are based upon equations I(463) to I(465) for the velocity gradient parameter and I(475) to I(478) for the standoff distance.

Next, the various heat flux values discussed in Section 8.1.4 of Volume I are computed. In preparation for these calculations, the dissociation enthalpy \( \Delta H_d \) of the gas at the inviscid stagnation point (denoted by HCF) is computed by summing the product of the mole fraction and the enthalpy of formation \( SHFA(I) \) over all neutral atomic species (see equation I(544)). In the case of an equilibrium shock, the mole fractions SAVEC(I) at the stagnation point are obtained from the common array ZCAP(I), which is set in subroutine EQCALC. In the case of a frozen shock, the mole fractions are the same as those in the free stream flow ahead of the shock, and are calculated from the concentrations \( GJ(I) \) using equation I(1). In the evaluation of HCF, the sum is restricted to neutral atoms by rejecting all species containing more than one element, more than one atom of an element, or a negative number of atoms of an element.* This algorithm

*Positive ions are represented as compounds containing a negative number of electrons.
accepts the electron species, which of course is not a neutral atom, but this does not affect HCF because the formation enthalpy of the electron is zero.

The transport properties $\mu_e$ (VISC), $N_{Pr}$ (PR), $N_{Le}$ (LE), and $\sigma$ (SIGMA) at the inviscid stagnation point are computed by calling subroutine TRANSP. The heat fluxes are calculated using equations I(466), I(468) and I(469) with suitable conversion factors to obtain English engineering units in the output. For example, the numerical coefficient in the Fortran formula corresponding to equation I(466) is obtained as follows:

$$0.51 = \frac{1.8 \times (30.48)^{3/2}}{453.6} \times 0.763$$

The factor of 1.8 converts the enthalpies from internal code units of cal/gm to Btu/lb; $(30.48)^{3/2}$ converts cm$^2$/cm$^2$ to ft$^2$/ft$^2$; and $(453.6)^{-1}$ converts g to lb.

After the heat fluxes have been computed, the results of the calculations are printed out. To this end, the results are first loaded into the array SCOUT, to allow use of a relatively compact output format. If a binary output tape is being prepared during the run (DATAPE = .TRUE.), certain of the model condition results are also written on tape unit ITPOUT.

### 3.31.2 Normal Shock Solution

Figure 15 diagrams the method used in the normal shock solution. This technique is explained in analytical terms in Section 8.1.1 of Volume I. The solution is based upon an iteration to determine the temperature $T_2$ behind the shock. The value of $T_2$ obtained during the previous call to MODEL is used as a first estimate; this value is stored as x2E (equilibrium shock) or T2F (frozen shock) in common block /STAG/. Prior to the first call to MODEL, these values are initialized to the reservoir temperature.

The criterion for selection of $T_2$'s that the static enthalpy behind the shock, calculated from $T_2$ using equation
Initialize EPS, T2

ICOUNT = 0
LΦ = .FALSE.
HI = .FALSE.

ISOLN = 1?

Equilibrium Shock

P2 = P1 + ρ1u1^2(1 - EPS)

ICOUNT = ICOUNT + 1

ICOUNT > 30?

yes

CALL EQCALC(T2, P2)
EPSOLD = EPS
EPS = ρ1/ZRHØ

yes

no

Recalculate P2

no

Error Msg.
EXIT

yes

ICOUNT > 30?

no

EXIT from Normal Shock Solution

F

H2 = HS - HD - EPS^2

LΦ = T2
TLØ = T2
T2 = 1.1 T2

ERLØ = ERZ
ERZ = ZCH - H2

ERZ ≤ 0?

yes

ERZ ≥ 0?

no

Compute T2 by Interpolation

T

HI = TRUE.
HI = TRUE.

Frozen Shock

CALL THERM

Compute ZCH, P2, R2, EPS

CT = T2/CTAP

CT=T2/CTAP

CALL THERM

Compute ZCH, P2, R2, EPS

Exit

Original page is of poor quality.
be equal to the value $H_2$ obtained from the Rankine-Hugoniot energy equation for the shock, equation (432b).

In the case of a frozen shock, the gas is thermally perfect because the mole fractions are constant. The static enthalpy $Z_{CH}$ is calculated, in this case, from species enthalpies obtained by calling subroutine THERM after the nondimensional temperature $CT$ has been set to the value $T_2/CTAP$. The pressure $P_2$ behind the shock is then computed from (437), and the shock density ratio from (434).

The equilibrium shock solution is more complicated because in this case the density ratio $\epsilon$ is not given as an explicit function of $T_2$. Thus, a double iteration is required. For each trial value of $T_2$, $\epsilon$ and the pressure $P_2$ are calculated by an inner iteration as described in the discussion around equation (438). In this case, the static enthalpy $Z_{CH}$ and density $Z_{RE}$ behind the shock are obtained by calling subroutine EQCALC. The outer iteration to determine $T_2$ is the same as for the case of a frozen shock.

The object of the outer iteration is to make $ERZ = Z_{CH} - H_2$ equal to zero, to within an accuracy of 0.1 percent. The temperature $T_2$ satisfying this criterion is obtained by iterative linear interpolation:

$$
(T2)_{new} = (T2)_{old} - \frac{ERZ \cdot (THI-TL)_{old}}{ERRHI-ERRL_{old}}
$$

(114)

Here $(T2)_{new}$ is the improved estimate resulting from the linear interpolation, $(T2)_{old}$ is the previous estimate, ERZ is the enthalpy error obtained at $(T2)_{old}$, ERRHI is the last positive error: ERZ obtained and THI the corresponding $T_2$ estimate, and ERRL is the last negative error obtained with TL$_0$ the corresponding temperature. Of course, this formula cannot be used unless points with both positive and negative ERZ have been computed. This requirement is implemented with the aid of logical variables L0 and HI which are initialized to "FALSE". L0 is set to "TRUE" when a negative ERZ is obtained, and HI is set to "TRUE" when a positive ERZ is found. If ERZ is negative and HI is "false", T2 is increased by 10 percent. If ERZ is positive and L0 is "false", T2 is
decreased by 10 percent. The interpolation formula (114) is not used until both \( \text{L} \) and \( \text{H} \) are \( \text{TRUE} \).

3.31.3 Conditions at the Inviscid Stagnation Point

After the normal shock calculation has converged, the conditions at the inviscid stagnation point are computed as explained in Section 8.1.1 of Volume I. A flowchart of the coding is given in figure 16. The logic is similar to that of the normal shock solution (figure 15), but with two simplifications:

1. The stagnation pressure is calculated from the incompressible Bernoulli equation (428b) with no iteration. Thus, only a single iterative loop is required even for the case of equilibrium flow.

2. The initial estimate of the stagnation temperature \( \text{T}_S \) is taken to be the temperature \( \text{T}_2 \) just behind the normal shock. Since \( \text{T}_2 \) is known to be lower than \( \text{T}_S \), the logical variable \( \text{L} \) is not needed in the stagnation-condition solution. Linear interpolation is begun as soon as a positive \( \text{ERZ} \) is obtained, where \( \text{ERZ} \) is the difference between the known stagnation enthalpy \( \text{H}_S \) and the local enthalpy at the stagnation point \( \text{ZCH} \) computed from \( \text{T}_S \) and \( \text{P}_S \).

3.32 Subroutine NEWRAP

The function of subroutine NEWRAP is to compute the flow conditions at a point in the equilibrium flow solution, using the method explained in Section 6.2 of Volume I. First, the system of \( (c + 1) \) equations I(227) and I(252) is solved for the species mole fractions \( X_i \) and the pressure \( \text{P} \). Then the remaining flow variables are computed from these data.

The Fortran notation in NEWRAP is similar to that in EQCALC, already discussed in Section 3.18, but there are some differences and additions. \( \text{ISMCNR} \) denotes the number of dependent species. Initially, this is equal to \( n-c=\text{ISS-ISC} \).
Figure 16. Flowchart of the Solution for Conditions at the Inviscid Stagnation Point
The array CHII is redefined as the logarithm of the equilibrium constants I(225) for the dependent species:

\[ CHII(i-c) = -\frac{\mu_{i}^o}{R_0 T} + \sum_{j=i_m}^{c} \tilde{\nu}_{i-c,j} \frac{\mu_{j}^o}{R_0 T} \]  

(115)

where \( i_m = IM \) is the index of the first independent species, initially equal to 1. A new array PGJ is defined by

\[ PGJ(i) = \frac{S_i^o}{R_0} - CRRB \cdot W_i \]  

(116)

where \( S_i^o \) denotes the molar entropy of the \( i \)th species at the standard pressure, \( W_i \) is the molecular weight of the \( i \)th species, and* \n
\[ CRRB = \frac{S_0}{R_0 W_0} \]  

(117)

where \( S_0/R_0 \) is the nondimensional molar entropy of the gas in the reservoir and \( W_0 \) the reservoir molecular weight.

Immediately upon entry, NEWRAP calls subroutine THERM to compute the species thermal properties at the temperature \( CT \cdot T_0 \) corresponding to the current value of the nondimensional temperature \( CT \). NEWRAP then sets up the arrays CHII and PGJ, defined above, and obtains a set of initial estimates of the species mole fractions \( X_i = CAPX(i) \) from equation I(1):

\[ CAPX(i) = W \gamma_i \]  

(118)

The molecular weight \( W = CM \) and species concentrations \( \gamma_i = GJ(i) \) are the values in storage. For the first call to NEWRAP, these are reservoir values. In subsequent calls, they are the final values for the preceding flow point. Also, *CRRB is set in subroutine INGAS following the entry INTA.
SKIL(I) is initialized to CAPX(I), and ZP to \( \ln(PRES) \), where PRES is the stored value of the nondimensional pressure.

The Newton-Raphson iterative loop starts at statement 70. The system of linear equations to be solved in each step of the iteration is \( I(258) \). The constant terms \( F_j(Y_k) \) are loaded into the \( M2^{\text{th}} \) column of the matrix \( AA \), where \( M2 = c + 2 \). For \( j = 1 \) to \( c \), these terms are given by equation \( I(227) \), and are computed in the \( D\$ \) loops ending at statement 90. The constant term for \( j = c + 1 \) is given by \( I(252) \), and is loaded into \( AA(M1,M2) \) by the statements between 90 and 100. In the loop "D\$ 100", the upper limit ISSNR is the current number of species included, initially equal to ISS.

The matrix of coefficients for the system of equations \( I(258) \) is loaded into the first \( (c + 1) \) rows and columns of \( AA \). The coefficients \( I(233b) \) for \( j = i_m \) to \( c \) and \( k = i_m \) to \( c \) go into the first \( c \) rows and columns, and are set by the nested \( D\$ \) loops ending with statement 150. The coefficients \( I(261b) \) for \( j = c + 1 \) and \( k = i_m \) to \( c \) are computed in the \( D\$ \) loops ending at statement 170. The coefficient \( I(261c) \) for \( j = c + 1 \) and \( k = c + 1 \) is set up by the loop \( D\$ 180 \). Finally, the coefficients \( I(261a) \) for \( j = i_m \) to \( c \) and \( k = c + 1 \) are loaded into \( AA \) by the \( D\$ \) loops ending at statement 210.

If the number of dependent species ISMCNR is zero a logical indicator NDEPS is set to .TRUE. at the beginning of NEWRAP. Tests on NDEPS then cause the subroutine to skip all of the \( D\$ \) loops with \( D\$ \) indices running from 1 to ISMCNR, in which sums from \( c + 1 \) to \( n \) are evaluated.

During an equilibrium expansion to high Mach numbers, the mole fractions of ionized species can become extremely low as the gas temperature falls to a small fraction of its value in the reservoir. To avoid computational problems arising when the electron mole fraction becomes exceedingly small, the electrons and all ionized species are eliminated from the equilibrium calculation whenever the electron mole fraction drops below \( 10^{-20} \). The test for this condition is in statement 220. If \( CAPX(1) \) is found to be \( \leq 10^{-20} \), the following
actions are taken:

(1) The indicator JJK is reset from its initial value of 0 to 1. The latter value signifies that the elimination of charged species has already been performed.

(2) The number of ions (IC) is subtracted from ISMCNR and ISSNR.

(3) IM is increased from 1 to 2.

(4) AA(1,1) is set to 1, and all the other entries in the first row and first column of AA are set to 0.

(5) The electron mole fraction, CAPX(1), is set to 0.

(6) The mole fractions of all the ion species are set to 0.

Regardless of whether the charged species have been eliminated, the system of equations I(258) is solved by calling subroutine DSMSPL. Improved values for the species mole fractions and the pressure are then obtained using equations I(262). The correction factor \(1 + h_n^Z\) in I(262) is denoted by \(ZB\) for \(n = 1\) to \(c\) (i.e., for the mole fractions), and by \(ZC\) for \(n = c + 1\) (the pressure). If any of these correction factors turns out to be 0 or negative, the corresponding quantity (mole fraction or pressure) is adjusted by dividing it by 2. Based on the improved values of \(p\) and the \(X_1\), the other quantities in the calculation (\(ZP = \ln p/p_0\) and \(CGMU(K) = \ln X_K\)), are then recomputed. Next, the mole fractions for the dependent species are computed from the equilibrium relations I(224), which are written in the logarithmic form

\[
\ln X_i = \ln K_{pi} + (\nu^*_{i-c} - 1) \ln p + \sum_{\ell=1}^{c} \nu_{i-c,\ell} X_{\ell} \quad (119)
\]
The term $\ln K_{ij}$ is denoted in NEWRAP by CHII(i), equation (115). The $\ln X_i$ are computed in the array SKIL(i-c) and the resulting values are saved in CGMU(i). Then the $X_i$ themselves are computed by taking exponentials and are stored in SKIL(i-c) and in CAPX(i).

The convergence criterion for the iteration is that the absolute magnitudes of all of the relative corrections $h_n$ be smaller than or equal to $TEST = 10^{-6}$. When convergence has been achieved, the other flow variables are computed from the pressure, the mole fractions, and the known reservoir conditions. The relations used in these calculations are equations I(236), I(237), and I(242) for the molecular weight, density, and specific enthalpy, equations I(79b) and I(81) for the effects of gas imperfections, equation I(263) for the flow velocity, and equation I(264) for the mass flux. As noted in Section 3.22.1, the flow variables are computed in nondimensional form.

3.33 Subroutine NEXTMP

The function of subroutine NEXTMP is to determine the location (XMODEL) of the next point at which model condition calculations are to be done. In the case of channel flow solutions, the routine determines the location corresponding to the next specified channel width.

The first two arguments of NEXTMP specify which of the model points called for in the input have already been passed in the current solution. ITS is the index of the next of the specified test-section diameters or channel widths, while XM0D1 is the x-coordinate the next model point in the geometric sequence of model points.

In the case of channel flow (NPRFLS = 2), the geometric sequence is not used. NEXTMP simply calls subroutine FINDXC to determine the position XM0D2 at which the width of the wide (MBLth) profile has the specified value TCAR(ITS), and sets XM0DEL equal to XM0D2.

In the case of nozzle flow (NPRFLS = 1), the next model point may be either the next point (XM0D1) in the geometric sequence or the point with the next specified area
ratio. In this case, NEXTMP calls FINDX to determine the location (XMØD2) of the point with the area ratio TSAR(ITS), and sets XMØDEL to the lower of the two values XMØD1, XMØD2. If the next point turns out to be at the specified area ratio, ITS is incremented by 1 to signify that TSAR(ITS) has been used. If the next point is a point in the geometric sequence, XMØD1 is replaced by FACMP times XMØD1, which is the next subsequent value of the sequence.

3.34 Subroutine NØNEQ

NØNEQ is the controlling routine for the nonequilibrium flow solution. The overall method of solution has been explained in Section 7.2 of Volume I. Briefly, the solution is started upstream of the throat, near the reservoir, using the perturbation method. At each step in the perturbation solution, the quantities $\delta \gamma_i$ (which are measures of the departures of the reactions from equilibrium) are calculated and tested. When any one of these quantities reaches a specified size, the perturbation technique is abandoned and the solution is continued by numerical integration. If the integration is begun well downstream of the throat, it is carried out using the normal, direct method in which the specified nozzle geometry is taken to be a condition on the flow. However, if the integration is started upstream of the throat, an inverse method is used, in which the equilibrium density distribution $\rho(x)$ is used to define the flow. In this case, a switch is made from the inverse method to the direct method when the solution reaches a point slightly beyond the throat.

Figure 17 is an overall flowchart of subroutine NØNEQ. This flowchart shows how the various methods are organized to produce the entire solution. The operations enclosed in double boxes are represented by separate flowcharts and discussed below. IUPD is an indicator which is initialized to 1 (in subroutine READ), and is reset to 0 (in subroutine THRØAT) at the switch from the upstream to the downstream region. INEQ is another indicator, initialized to 0, and set to 1 when the numerical integration is started. Figure 17 does not show any exits from the control loops that generate the solution, because the case termination controls are within the boxes labelled "Numerical integration" and "Perturbation solution". The statements involving MØDLPT, XMSET, and subroutine NEXTMP pertain to the calculation of conditions at model points, and are discussed below.
Figure 17. Flowchart of Subroutine NSEQ
The remaining parts of this section discuss individual portions and aspects of subroutine N\textup{ONEQ} in detail.

3.34.1 Initializations

Figure 18 is a flowchart of the initializations section of N\textup{ONEQ}, represented by the uppermost double box in figure 17. This flowchart shows only the initializations of control variables. In the subroutine, several physical variables are also set, but their values are not needed for following the logic.

The perturbation technique, which is used to start the solution, proceeds by taking steps in temperature, \( \Delta T \). The initial temperature decrement is given by \( \Delta T/T_0 = \text{DELT1} \), where the variable \( \text{DELT1} \) is under input control and has a preset value of 0.01. This preset value of \( \text{DELT1} \) gives fairly large steps in the perturbation solution. In cases including the boundary layer (\( \text{ISW3B} \neq 0 \)), to allow more accurate evaluation of the boundary layer integral \( I(172) \), \( \text{DELT1} \) is reset to 0.049 \((1 - \text{CTMAX})\), where \( \text{CTMAX} \) is the nondimensional sonic temperature \( T*/T_0 \) based on the equilibrium solution. This value gives 20 steps upstream of the throat if the perturbation technique is used all the way to the throat. If the numerical integration is started upstream of the throat, as is usually the case, the number of steps is larger because the initial step size in the integration is much smaller than the initial perturbation step size.

The temperature, species concentrations, and gas properties are initialized for the nonequilibrium solution in the main program. Before calling N\textup{ONEQ}, the main program sets the nondimensional temperature \( \text{CT} \) equal to \((1 - \text{DELT})\), where \( \text{DELT} \) is set by the same rules used in calculating \( \text{DELT1} \) in N\textup{ONEQ}. MAIN then calls NEWRAP to determine the equilibrium flow conditions at this temperature, calculates the effective area ratio, and determines the corresponding position in the nozzle by calling FINDX.

As explained in Section 7.3 of Volume I, the system of equations used in the perturbation technique is solvable only if the \( \beta \) matrix has a rank of \( n-c \). The initializations section of N\textup{ONEQ} computes the rank \( IX \) of \( \beta(I,J) \) and executes an error exit if \( IX \) is not equal to \( n-c \).
Figure 18. Flowchart of Initializations Section of Subroutine NØNEQ
3.34.2 Switch to Downstream Region

Figure 19 is a flowchart of the operations represented by the box labelled "Switch to downstream region?" in figure 17. The coding shown in figure 19 determines whether the conditions for switching to the downstream solution have been satisfied, and if so, carries out the switch by calling subroutine THR3AT. The conditions are as follows:

1. \( DL\theta GA = \frac{d \ln A}{dx} \), where \( A \) is the area ratio computed using the inverse method, must be nonnegative.

2. The position of the flow point must lie beyond the geometric throat \((CX > 0)\).

3. The area ratio \( \tilde{A} = \frac{AFNX}{ANX} \) must be greater than or equal to a value \( DATEST \) which is preset (in subroutine INIT) to 1.01.

When all of these conditions have been satisfied, a large number of flow variable and control parameters are saved in the arrays TB, ITB, and BLBK, and the switch to the downstream region is executed by calling THR3AT. In THR3AT, IUPD is reset to 0 and RSA is set to a value which rescales the actual nozzle geometry to make it consistent with \( \tilde{A} \).

In cases including the boundary layer, the coding in figure 19 also sets the PMST(I) array based on the displacement thickness DEBL(I) at the first point where the conditions (1) and (2) are satisfied.

3.34.3 Restart of the Upstream Solution

Figure 20 is a flowchart of the operations represented by the box labelled "IUPD = 1, Increase DATEST, Restart at Switch Point" in figure 17. This section of NGNEQ is entered when a positive value of \( \frac{d \ln \rho}{dx} = DL\theta GR \) is encountered in the downstream solution. If this condition arises so far beyond the upstream/downstream switch point that the effective area ratio \( A_0 = AFNX \) exceeds \( DATEST \) by 5 percent of \( DATEST \) or more, an error exit is executed. Otherwise,
Figure 19. Flowchart of Switch to Downstream Region in Subroutine NØNEQ
Write "DL0GR IS POSITIVE"

\[
\frac{(AFNX-DATEST)}{DATEST} < 0.05? \quad \text{no \rightarrow Error Exit 3}
\]

\[
\text{DATEST} = 2 \times (DATEST-1) + 1.
IUPD = 1
ARBB = ARBB + 1.
\]

\[
\text{no \rightarrow Error Exit 4}
\]

\[
\text{yes \rightarrow ARBB < ARBA?}
\]

\[
\text{yes \rightarrow TSZ = TPRINT}
TPRINT = 0.
\]

\[
\text{CALL PRTA}
\]

\[
\text{ERR? \rightarrow T \rightarrow RETURN}
\]

Backspace binary output tape

TPRINT = TSZ

Reset data stored in TB, ITB, BLBK

Figure 20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine N0NEQ
the program assumes that the positive $d \ln \rho / dx$ value is a result of switching to the downstream solution too near the throat, and attempts to recover the correct solution. To this end, the excess of DATEST over unity is doubled, IUPD is reset to 1 (signifying the upstream solution), the flow variables and control parameters are all reset to their values at the previous switch point, and the solution is restarted at that point. A counter ARBB (initialized to 0, in subroutine INIT) is also incremented by 1. If the $d \ln \rho / dx > 0$ condition is encountered repeatedly after successive restarts, the effort to continue the solution is abandoned after four attempts (ARBB $\geq$ ARBA = 5., set in INIT). Also, before the restart is executed, the conditions at the final point of the invalid downstream solution are printed out by setting TPRINT = 0. and calling PRTA. If a binary output tape is being produced for subsequent use by the NATA plot program, the records containing data from the invalid solution beyond the switch point are eliminated by backspacing the tape.

3.34.4 Perturbation Solution and Switch to Integration

Figure 21 is a flowchart of the operations represented by the boxes labelled "Perturbation solution" and "Switch to integration?" in figure 17. These operations may be summarized as follows:

(1) Subroutine PERT is called to compute the perturbations PCT (in the nondimensional temperature), PERTG(J) (in the species concentrations), and SDCHI(I) (in the reaction parameters $\chi_i$).

(2) If the solution is in the downstream region (IUPD = 0) and the nondimensional temperature decrement DEFT is still equal to the reduced value of DEFT1 (see part 3.34.1 above), DEFT is reset to the input value of DEFT1, which was saved in DEFTSV. This is done because the step size in the perturbation solution is sufficiently small in the throat and near downstream region, even when the larger decrement is used.
Figure 21. Flowchart of Perturbation Solution and Switch to Integration in Subroutine H333EQ
(3) The largest \(SDCHI(I)\) value \(DCHMAX\) and the smallest \(DCHMIN\) are determined.

(4) The ratio \(RATI\phi = DCHMIN/DCHMAX\) is examined. If this ratio is larger than or equal to a value \(DCHRAT\) (preset in BLOCK DATA to \(10^{-4}\)), \(N\!O\!N\!E\!Q\) proceeds to the checks for the switch to numerical integration, item (5) below. If \(RATI\phi\) is less than \(DCHRAT\), the reaction system assumed in the gas model will lead to a switch to integration at a point where some of the reactions are still very near equilibrium. The value of \(RATI\phi\) is approximately independent of the size of \(DCHMAX\); thus, the program responds to an excessively small value of \(RATI\phi\) immediately, even though \(DCHMAX\) may be much smaller or much larger than the value \(CCHI\) at which the switch to integration is desired. To prevent the premature start of the numerical integration, \(N\!ATA\) artificially increases the rate constant for the reaction \((I = IMAX)\) for which \(|SDCHI(I)| = DCHMAX\), if this can be done without significantly affecting the solution. This procedure is discussed near the end of Section 7.3 in Volume I.

If \(RATI\phi < DCHRAT\), \(N\!O\!N\!E\!Q\) determines whether this reaction involves a minor species (concentration less than or equal to \(GAMIN\), preset to \(10^{-10}\)) whose concentration is being decreased by the reaction. If so, the rate constant for the \(IMAX^{th}\) reaction is increased by a factor of \(RATI\phi2 = 1.1*DCHRAT/RATI\phi\), and the perturbation calculation for the previously assumed temperature is repeated. If not, the perturbation solution is continued without adjusting the rate constant.

(5) The next step is to check whether \(DCHMAX\) has reached the value \(CCHI\) at which the switch to the numerical integration is desired. If \(DCHMAX\) is less than \(CCHI\), the perturbation solution continues. If the axial coordinate
CX is greater than or equal to the current value of XMODEL (the coordinate of the next model point), the indicator M0DLPT is set to .TRUE., and subroutine NEXTMP is called to determine a new value for XMODEL. In any event, subroutine PRTA is called to provide output of the computed flow conditions at the current point of the perturbation solution. If M0DLPT is .TRUE., PRTA also calls subroutine M0DEL to calculate test conditions on a model.

(6) After the return from PRTA, the case termination tests are applied. If CX is greater than or equal to CXMAX, or if the nondimensional temperature CT is less than or equal to TSTOP, control is transferred to statement 660. The flow conditions and model conditions at the final point are then printed and the RETURN is executed.

(7) If case termination has not been reached, the temperature CT is decremented by DELT, and subroutine NEWWRAP is called to compute the equilibrium flow conditions at the new temperature. The equilibrium effective area ratio AFNTS is then calculated from the continuity equation I(265), and subroutine AXFIT is called to determine the axial coordinate CX at this area ratio. Control is then transferred to statement 220 ("CALL DERIVS" in figure 17).

(8) Once DCHMAX reaches or exceeds the criterion value CCHI for the switch to integration in step (5), NJ0NEQ commences an iterative adjustment of the nondimensional temperature CT in order to obtain a flow point at which DCHMAX= |δX i| max satisfies both of the conditions in equation I(381). Usually the perturbation solution overshoots the second condition I(381), i.e., DCHMAX is greater than PCTEST*CCHI, where
PCTEST = 1.2 (set in subroutine INIT). When this occurs, NØNEQ cuts DELT in half, adds DELT to CT in order to obtain a temperature halfway between the current value and the preceding one, and computes the corresponding point in the perturbation flow solution by calling NEWRAP and AXFIT as in step (7) above. This procedure, together with the normal procedure for advancing the perturbation solution (step (7)), repeatedly subdivides the temperature interval known to contain the flow points satisfying the conditions I(381).

(9) When a point satisfying both of the conditions I(381) has been obtained, the switch from the perturbation technique to numerical integration is carried out. The indicator INEQ is reset from 0 to 1 and the perturbations PCT, PERTGJ(J) and PRHØ are added to the corresponding equilibrium flow quantities CT, GJ(J), and RHØ. The integration step size DELTAX is adjusted, if necessary, to make it no larger than DXM = 0.01* DCHMIN/DCHLL, where the input DCHLL is preset to 10^-4 in BLOCK DATA. This adjustment provides a reduced initial step size when DCHMIN = |\( \Delta X_i \)|_{\text{min}} at the switch point is smaller than 10^-4, i.e., when some of the reactions are still very close to equilibrium. If this reduction in DELTAX were omitted, the normal step size controls of the integration would still bring DELTAX down to similar values at the cost of some additional computation. Any rate constants which have been artificially increased in step (4) are now restored to their correct values. Subroutine DERIVS is called to compute the derivatives and supplementary flow variables at the switch point, based on the perturbed temperature and concentrations, and control is transferred to the numerical integration section of NØNEQ. If the indicator FAILED is .TRUE. after the call to DERIVS, an error exit is executed. This indicator is .TRUE. if the flow conditions fail to pass certain validity checks in DERIVS and COMM.
3.34.5 Numerical Integration

Figure 22 is a flowchart of the operations represented by the box labelled "Numerical integration" in figure 17. These operations may be summarized as follows:

(1) The case-termination checks are the same as in the perturbation solution. If \( CX \geq CX \text{MAX} \) or \( CT \leq TST\text{OP} \), control is transferred to statement 660. The flow conditions and model conditions at the final flow point are then printed, and the RETURN is executed.

(2) If the end of the case has not been reached, the counter NNN is incremented and subroutine PRTA is called to provide output of the conditions at the flow point reached during the preceding integration step.

(3) Next, the integration step size DELTAX and the factors SC and SCD used in changing DELT Y are adjusted. If NNN is equal to NQS (an input parameter, preset to 4), then DELTAX is multiplied by SC, NNS is incremented by 1, and NNN is reset to 0. Then if NNS is equal to NQS, SC is increased by 0.1, SCD is reset to the new SC value, and NNS is reset to zero. For NNN < NQS, no changes are made in these variables.

(4) In preparation for the next integration step, the current values for CX, CT, GJ(I), and other quantities are saved in the storage locations CXB, CTB, GJB(I), etc. If the forthcoming step should fail for any reason, these data will be used in restarting the solution at the beginning of the step with a smaller step size.

(5) Next, the anticipated position, \( CX + \text{DELTAX} \), at the end of the forthcoming step is compared with XMØDEL and CXMAX. If \( CX + \text{DELTAX} \) is greater than either of these values, DELTAX is reduced to make the end of the step coincide with the lesser of
Figure 22. Flowchart of Controls for the Numerical Integration in Subroutine N/CX
the two. If the end of the step will be at $X^{\text{MODEL}}$, then $\text{MODELPT}$ is set to $\text{.TRUE.}$ and the DELTAX value before the reduction is saved in DXSAVE.

(6) Subroutine RNKT is called to compute the changes in the dependent variables over the integration step.

(7) The indicator FAILED is tested. A $\text{.TRUE.}$ value of FAILED indicates that a validity check was not satisfied somewhere in RNKT, DERIVS, or $\text{COMM}$. If FAILED = $\text{.TRUE.}$, the step size is reduced by dividing DELTAX by SCD and the calculation is re-started at the beginning of the step, using the data saved in step (4). In the event of such a step restart, a counter $\text{COUNT}$ is incremented. Also, MODELPT is set to $\text{.FALSE.}$, SCD is increased by a factor of 1.1, and SC is reduced by 0.1 (but not below a value of 1.1). In addition, NNN and NNS are both set to zero. The idea underlying these changes in the step control parameters is that the failure was caused by instability resulting from the use of an excessively large step size. Either the step has been increased too rapidly, or the solution is entering a region where the reaction system requires a smaller step size for stability. Thus, the step size itself is reduced, and the growth in step size is both temporarily arrested and reduced in rate. The increase in SCD allows the program to reduce the step size very rapidly if repeated failures are encountered in the same step, as sometimes happens in runs using electronic nonequilibrium models.

(8) If FAILED is $\text{.FALSE.}$ after the call to RNKT, additional validity checks I(407) and I(408) are applied to the results of the integration step. If any of these conditions is violated, the step is restarted with a smaller step size as described above in item (7). Also an indicator IF\'IL is set to a distinctive value for each possible type
of failure. IFAIL is also set when step failures occur in other subroutines. Each time a step fails and is restarted, the IFAIL value is saved in the integer array IF(IC$UNT). These values can be obtained in the printed output by setting ISW5B to a negative value in the code input. Output of these data shows what tests, species, etc., are controlling the integration step size.* If a failure occurs in one of the conditions I(407a), I(407c) on the species concentrations, then the program checks whether the procedure of freezing the concentration of a minor species (Section 7.5.3 of Volume I) is applicable. The conditions for use of this procedure are that the species I, whose concentration $GJ(I)$ violates I(407a) or I(407c), should have a concentration $GJB(I)$ at the start of the step less than GAMIN, and that the concentration change for this species over the step, $SDGJ(I)$, be negative. If these conditions are satisfied, all of the reactions contributing to changes in the concentration of this species are frozen by setting the corresponding elements of the array NØREAC(K) to .TRUE.. The step is then restarted in the usual way.

(9) If the conditions I(407) and I(408a,b) are satisfied, the integration step is presumed to be valid subject to the condition I(408c), which is applied later. The changes SDT, SDTE, SDCHA in the dependent variables during the step are added to the values CTB, CTEB, CHB at the start of the step to obtain the corresponding values at the end of the step. The species concentrations are then adjusted to restore the original elemental composition of the gas mixture, as described in Section 7.1.3 of Volume I and in Section 3.34.6 below. Subroutine DERIVS is then called to compute the derivatives and supplementary flow variables at the end of the step.

*The numerical code used in IF(IC$UNT) is defined in Section 4.55.
(10) If \( NT = 2 \) (electronic nonequilibrium model), the condition \( I(408c) \) on the change in the energy transfer to the electron gas (QDPE or \( q_e \)) is now applied to the final QDPE value resulting from this call to DERIVS. If this condition is not satisfied, the step is restarted with a reduced step size in the usual way.

(11) Once all of the validity checks have been passed, SCD is reset equal to SC and the step-restart counter \( ICOUNT \) is reinitialized to 0. The counter \( ISTEPS \) for successful steps is incremented by 1. The step size \( DELTAX \) is set equal to the larger of \( DELTAX \) itself and \( 0.7*DXOLD \). Thus, if \( DELTAX \) has been reduced sharply by repeated step restarts, it is here returned to 70 percent of its value in the preceding step.

(12) Control is now transferred to statement 210 ("MOLDLPT?" in figure 17).

### 3.34.6 Element Conservation

The adjustment of species concentrations to restore conservation of the chemical elements, explained in Section 7.1.3 of Volume I, is implemented by the coding from statement 800 down to 940 in \( N\)\( N\)\( O\)\( N\). The programming is straightforward, and can easily be related to the formulas in Section 7.1.3 with the aid of the following summary of notations:

- \( CJ \)
- \( DCI(J) \)
- \( GSQ(I) \)
- \( AIN(J,K) \)
- \( DCA(K) \)
- \( AA(I,ISMCP1) \)

\[
\begin{align*}
CJ & \quad c_j' \\
DCI(J) & \quad \delta c_j \\
GSQ(I) & \quad (\gamma_i')^2 \\
AIN(J,K) & \quad A_{jk} \\
DCA(K) & \quad \frac{1}{(\gamma_k')^2} \sum_{j=1}^{c_j} \delta c_j \cdot A_{jk} \\
AA(I,ISMCP1) & \quad b_i \text{ (before call to DSMS@L)}
\end{align*}
\]
The system of equations I(346) is solved for $\gamma_m$ by calling subroutine DSMSPL. After the adjusted concentrations of the dependent species ($i = c + 1$ to $n$) have been calculated from I(340) and those of the independent species ($i = 1$ to $c$) from I(338), the condition I(407a) is applied to the adjusted concentrations. If any of these concentrations is found to be negative, the step is restarted with a reduced step size in the usual way.

3.34.7 Model Points

The position of the next model point, at any point in the flow, is denoted by XMODEL. Whenever a particular model point has been reached and the model calculations have been done, the new XMODEL value is obtained by calling subroutine NEXTMP(ITS,XMÐ1,XMÐ1). The argument ITS is the index of the next specified test section diameter, and XMÐ1 is the coordinate of the next model point in the geometric sequence of model points. Subroutine NEXTMP, each time it is called, determines whether the next model point is at the test section diameter with index ITS or at XMÐ1, sets XMÐ1 accordingly, and updates ITS or XMÐ1 as required. Before NEXTMP is called for the first time in N0EQ, the initial values ITS = 1 and XMÐ1 = XMP1 must be set. This is done by coding shown in figure 17, at the first point where the Mach number AMACH has reached or exceeded a value of 1.5. A .TRUE. value of the indicator XMSET denotes that this initialization has already been performed.

The indicator M0DLPT, if .TRUE., denotes that the current flow point in the perturbation solution or the integration is a model point. If M0DLPT = .TRUE., PTA calls entry OUT2 of subroutine OUT1 to print the flow conditions (even though the TPRINT criterion may not be satisfied) and calls MODEL to compute and print the model test conditions.
When a model point is encountered during the numerical integration, DELTAX is reset to provide a flow point exactly at the model point. During the perturbation solution, MODEL is called at the first solution point beyond XMODEL. To obtain a perturbation solution point coinciding with XMODEL would require an iteration, as in the equilibrium and frozen flow solutions, because the flow is generated by decrementing the temperature, not by incrementing the position coordinate. Such an iteration has not been provided in the coding because, in the normal applications of NATA, the switch from the perturbation technique to the numerical integration always occurs upstream of the point where the Mach number is equal to 1.5.

3.34.8 Error Exits

Subroutine NQNEQ contains ten error exits, marked conspicuously with comment cards in the listing. Whenever one of these exits is executed, a message giving the error exit number is written out before the DUMP routine is called to provide diagnostic data. This message facilitates determining the nature of the error that caused NQNEQ to fail.

3.35 Subroutine NRMAX

This subroutine determines the sonic flow conditions for the case of equilibrium flow. Its logic and structure are practically identical with those for the corresponding calculation for frozen flow, performed in subroutine FRZEQ. The flowchart for the frozen calculation (figure 9) is applicable to subroutine NRMAX. The discussion in Section 3.22.2 is also applicable. In the present case, the mass flux at each temperature is determined by calling subroutine NEWRAP.

3.36 Subroutine OUT

This routine simply prints a list of definitions of the labels used in the output of the flow conditions and test conditions on models and wedges.
3.37 Subroutine \$UT1

Subroutine \$UT1 is the principal output routine of \$ATA. When called through its first entry point, it produces a printed list of the conditions at a special point in the flow, such as the reservoir or the sonic point. When called through its entry \$UT2(ISOLN), it prints a more comprehensive specification of the conditions at a general flow point. In addition, if the indicator DATAPE is \$.TRUE., a call to \$UT2 causes a record to be written on the binary output tape (tape 8), for subsequent use by the \$ATA plot program. Figure 23 is an overall flowchart of the subroutine.

3.37.1 Entry \$UT1

\$UT1 is called (by MAIN) to print the conditions in the reservoir and at the frozen and equilibrium sonic points. Since a call to \$UT1 precedes each of the three flow solutions, \$UT1 is also used to initialize some quantities used by \$UT2.

The chemically frozen specific heat ratio GAMMA is computed in a section of coding which is used by both \$UT1 and \$UT2. First, the entry THERML of subroutine THERM is called to compute the nondimensional species specific heats \(C_{pi}/R_0\) = CCPJ(I) at the current nondimensional temperature \(CT = T/T_0\). The nondimensional molar specific heat of the gas mixture is then computed as

\[
C_P = \frac{C_P}{R_0} = \sum_{i=1}^{n} X_i \frac{C_{pi}}{R_0} \quad (120)
\]

where the species mole fractions \(X_i\) are obtained from the array SAVEC(I), where they have been placed by the calling routine. Then the thermodynamic relation

\[
C_P - C_V = R_0 \quad (121)
\]

is used to obtain the specific heat ratio, \(\gamma\):

\[
\gamma = \frac{C_P}{C_V} = \frac{1}{1 - \frac{R_0}{C_P}} \quad (122)
\]
ENTRY \texttt{OUT1}

IFLAG=1
Initializations
IS6=IABS(ISW6B)
I6=0

CALL THERM1

Compute CP
Compute GAMMA

IFLAG?

1
Set output variables
P\texttt{OUT}, U\texttt{OUT}, etc.

Compute FAREA
Compute TFL\texttt{OW}

Print Output

RETURN

2

Print flow conditions at a point in the solution

RETURN

Figure 23. Overall Flowchart of Subroutine \texttt{OUT1}
At the conclusion of this calculation, the indicator IFLAG is used to transfer control to the appropriate portion of the subroutine, depending on which entry point was called. If entry was through OUT1, control passes to statement 30. The variables used for output of the flow conditions in OUT1, namely POUT, UOUT, etc., are then set. Next, the effective cross sectional area of the flow at the throat, FAREA, is computed. For ISW3B = 0 (boundary layer neglected), FAREA is based upon the geometric dimensions of the nozzle or channel at the throat. For ISW3B ≠ 0, the displacement thickness at the throat is taken into account using equation I(125), I(129), or I(133) as appropriate. After it has been calculated in square centimeters, FAREA is converted into square feet. The total mass flow is then computed by multiplying FAREA by the sonic mass flux in lbm/ft²sec.

Finally, the flow conditions, including the calculated mass flow TFLON, are printed out in the form of a list. This is followed by a list of species mole fraction.

3.37.2 Entry OUT2

The operations performed when the entry OUT2(ISOLN) is called, which are represented by a double box in figure 23, are flowcharted in detail in figure 24. The complexity of this portion of the subroutine results from the need to deal with various special cases:

- **ISW3B ≠ 0**: Boundary layer included; output includes boundary layer displacement and momentum thicknesses, heat flux, shear stress, Stanton number, etc.
- **ISW3B = 0**: Boundary layer neglected.
- **NPRFLS = 1**: Flow is in a nozzle.
- **NPRFLS = 2**: Flow is in a channel; two full sets of boundary layer outputs if ISW3B ≠ 0.
- **JDIM = 0**: Two dimensional nozzle.
Figure 24a. Flowchart of the Operations Produced by Calling Entry $GUT2$ of Subroutine $GUT1$ (Part a)
Figure 26b. Flowchart of the Operations Produced by Calling Entry SUM of Subroutine SUM (Part b)
JDIM = 1     Axisymmetric nozzle.

LIMOUT ≠ 0     Include output of SN, XSN boundary layer parameters.

LIMOUT = 0     Omit this output.

The printed outputs are, in most cases, expressed in different physical units from those used in the internal computations of the code. Thus, a major function of $\text{OUT2}$ is to re-express these quantities in the desired output units. The converted flow variables in output units are loaded into an array $\text{FVOUT}(I)$, dimensioned (35). This scheme allows the use of compact WRITE statements based on implied DO loops. The converted quantities in $\text{FVOUT}$ are also used in the calculations of conditions on wedge models; they are communicated to subroutine WEDGE through the common block $\text{OUTPUT}$. Definitions of the $\text{FVOUT}$ array elements are given in the glossary of Fortran symbols, Section 4.31. However, the definitions will be reviewed here to facilitate the study of the flowchart (figure 24).

The following elements of $\text{FVOUT}$ always have the meanings indicated:

1. Axial coordinate, $x$ (inches)
2. Temperature, $T$ ($^\circ$K)
3. Enthalpy, $h$ (Btu/lb)
4. Pressure, $p$ (atm)
5. Density, $\rho$ (lbm/ft$^3$)
6. Velocity, $u$ (ft/sec)
7. Mach number, $M$
8. Entropy, $s$ (Btu/lbm$^{-\circ}$R)
9. Frozen specific heat ratio, $\gamma$

\[-145-\]
(11) Effective area ratio, $A_e$
(12) Reynolds number per foot
(13) Molecular weight (g/mole)
(14) Viscosity (lbf/ft·sec)
(15) Electrical conductivity (mho/cm)

Element number 6 is absent from this list because its meaning varies with the type of case, as explained below. For ISW3B ≠ 0, the Reynolds number per foot, viscosity, and electrical conductivity are calculated in BLAYER, and $\phi_{UT2}$ obtains these data from common blocks /BLOUT/ and /TRPR@P/. For ISW3B = 0, BLAYER is not called and these quantities must be computed in $\phi_{UT2}$. The viscosity and conductivity are obtained by calling subroutine TRANSP.

If the boundary layer is included (ISW3B ≠ 0), the ten elements $FV@UT(16)$ to $FV@UT(25)$ are loaded with data relating to the boundary layer. If, in addition, a channel geometry is being used (NPRFLS = 2), these elements contain the data for the boundary layer on the first profile, and the ten elements $FV@UT(26)$ to $FV@UT(35)$ contain the corresponding data for the second profile. The meanings for the elements numbered 17 to 25 are as follows:

(17) Boundary layer displacement thickness (inch)
(18) Boundary layer momentum thickness (inch)
(19) Heat flux to nozzle wall (Btu/ft$^2$·sec)
(20) Shear stress on nozzle wall (lbf/ft$^2$)
(21) Recovery enthalpy (Btu/lbm)
(22) Prandtl number at the reference temperature
(23) Stanton number
(24) Reynolds number $Re_\delta$ based on momentum thickness

(25) Value of $Re_\delta$ for boundary layer transition

The data on geometric area ratio and nozzle diameter or channel widths is placed into $FV\thetaUT(6)$, $FV\thetaUT(16)$, and $FV\thetaUT(26)$. The arrangement of these data depends upon the type of case being treated:

A. If $ISW3B = 0$, $NPRFLS = 1$: $FV\thetaUT(6) =$ nozzle diameter (inch). In this case, the geometric area ratio is equal to the effective area ratio, $FV\thetaUT(11)$.

B. If $ISW3B = 0$, $NPRFLS = 2$: $FV\thetaUT(6)$ is set to a large value which exceeds the width of the format used and thus prints as a row of asterisks. The width of the channel surface lying on the first profile is set to $CHDIMS(2)$. The corresponding width for the surface with the second profile goes into $CHDIMS(1)$. These $CHDIMS(I)$ values are printed on a separate line in the block of output. The geometric area ratio is again equal to the effective area ratio.

C. If $ISW3B \neq 0$, $NPRFLS = 1$: $FV\thetaUT(16) =$ geometric area ratio, $FV\thetaUT(6) =$ nozzle diameter (inch).

D. If $ISW3B \neq 0$, $NPRFLS = 2$: $FV\thetaUT(6) =$ geometric area ratio; $FV\thetaUT(16) =$ width of channel surface lying on the first profile (inch); and $FV\thetaUT(26) =$ width of channel surface lying on the second profile (inch).

Note that in the channel case ($NPRFLS = 2$), the width of the surface lying on a given profile is twice the ordinate of the other profile.

After the data have been loaded into the output arrays, they are printed out. First, a line of asterisks is printed to separate the new block of output from the previous one. Imbedded in this line is a word ("frozen", "equilibrium", or "nonequilibrium") identifying the type of solution being
computed, based upon the entry-point argument \textit{IS\$LN}. In the case of a nonequilibrium solution, the number of integration steps since the last output of flow variables (NSTEPS) and the value of the indicator INEQ are also printed in this divider line. Note that NSTEPS is initialized in OUT1 to the seven-digit value 1000000, so that when printed (with an I6 format) in the frozen and equilibrium solutions, it gives a row of six asterisks.

Following the divider line, the first fifteen elements of FVOUT(I) are printed by a single WRITE statement. The array VARNAM(I) appearing in this statement contains the Hollerith labels for the output variables. Then, if ISW3B is nonzero, the boundary layer data in FVOUT(16) to FVOUT(25) or FVOUT(35) are printed. For channel cases without the boundary layer, the channel widths CHDIMS(I) are printed on a separate line. If LIMOUT is nonzero, the boundary layer parameters stored in FVOUT(I,J) are printed on a separate line. For nonequilibrium solutions based on an electronic nonequilibrium model, the electron temperature, the radiative and electronic energy transfer terms, and the total enthalpy are loaded into TNOUT(I) and printed.

Next, if ISW6B is nonzero and IS\$LN is not equal to 1, and if I6 is equal to IS6, the species mole fractions are printed. The test on I6 provides output of these species data every IS6th time OUT2 is called in the equilibrium and nonequilibrium solutions, where IS6 = |ISW6B|. Finally, if ISW6B is negative and IS\$LN = 3 (nonequilibrium solution) and if I6 = IS6, a table of reaction rate data is printed out. This table includes $P_i$, $\gamma_i$, and $P_i \gamma_i$ for all of the reactions ($i = 1$ to $r$), and $d H_i / dx$ for all the species ($j = 1$ to $n$).

If DATAPE is .TRUE., a record is written on tape 8 for subsequent use by the \textit{DATA} plot program.

3.38 Subroutine \textit{PERT}

The function of subroutine \textit{PERT} is to compute the perturbations in the flow variables by solving the following system of $n+2$ linear equations:
The unknowns are \( \text{PERTGJ}(J) = \delta \gamma_j \) for \( j = 1 \) to \( n \), \( \text{PCT} = \delta T / T_0 \) and \( \text{PRH} = \delta \rho / \rho_0 \), in the order listed. The simultaneous solution is obtained by calling subroutine DSMS\\( \varnothing \)L.

The coefficients and constants for the above system of equations are set up in the common array AA. First, the region of AA which is to be used is zeroed. Then, in the D\\( \varnothing \) loops ending at statement 20, the coefficients for equations I(368) are set into AA(I,J) for I = 1 to ISC. The coding down to the statement following 110 then loads the coefficients of equations I(362) into AA(I,J) for I = ISC + 1 to ISS, and loads the coefficients for equation I(374) into AA(ISSP1,J). In the calculation of the coefficients for I(362), the Fortran index I corresponds to \( j \) in I(362), \( J \) corresponds to \( k \) and \( K \) corresponds to \( i \). The coding from D\\( \varnothing \) 120 down to the second statement below 130 sets up the coefficients for equation I(380) in AA(ISSP2,I). Finally, the loop D\\( \varnothing \) 140 loads the constant terms of equation I(362) into AA(I,ISSP3) for I = ISCPL to ISS. The remaining equations are all homogeneous, so that their constant terms have been set by the initial zeroing of the AA array.

After DSMS\\( \varnothing \)L has been called, the values obtained for the perturbations are retrieved from the column AA(I,ISSP3). Then the \( \delta \kappa_i \) values, denoted by SDCHI(I), are calculated from equation I(360).

### 3.39 Function PI\\( \varnothing \)MEG

This routine provides the Maxwell-averaged momentum transfer cross section for collisions of electrons with
neutral species as a function of the electron temperature, TE. The cross section is calculated by linear interpolation in a table stored in common block /TNEQ/.

3.40. **Subroutine PRØP**

The function of this routine is to compute the flow conditions at a specified temperature in the frozen flow solution. Since PRØP makes use of species thermal properties, a call to PRØP must always be preceded by a call to THERM.

The technique used by PRØP has been discussed in Section 6.4 of Volume I. The coding is straightforward. As noted in Section 3.22.1, the flow variables are computed in nondimensional form. In particular, equation I(267) for the pressure is written in the form

\[
Z_P = \ln \left( \frac{P}{P_0} \right) = -\ln P_0 + \sum_{j=1}^{n} X_{j0} \frac{S_j^o}{R_0} - \sum_{j=1}^{n} X_{j0} \gamma_j X_{j0} - \frac{W_0 S_0}{R_0}
\]  

(123)

The first sum on the right is evaluated in PRØP as

\[
\sum_{j=1}^{n} X_{j0} \frac{S_j^o}{R_0} = W_0 \sum_{j=1}^{n} \gamma_j S_j^o = CMA - S4
\]  

(124)

The remaining three terms on the right in (123) are computed as \( W_0 \cdot ENT \), where \( ENT \) has been calculated in PRØZEQ before PRØP is first called:

\[
ENT = -\sum_{j=1}^{n} \gamma_j \left( \ln \gamma_j + \ln W_0 \right) - \frac{\ln P_0}{W_0} - \frac{S_0}{R_0}
\]  

(125)

3.41 **Subroutine PRTA**

This subroutine controls the output of the flow conditions and executes the calls to the MØDEL routine during
the nonequilibrium solution. It also calls subroutine BLCALL with the argument FINAL = .TRUE., to update the derivative of the boundary layer displacement thickness and provide permanent increments to the Cohen-Reshotko boundary layer integral. PRTA calls BLCALL even if the boundary layer is being neglected in the flow solution (ISW3B = 0), because the output and model routines need the species mole fractions, which are computed and loaded into common block /TEMPRY/ by BLCALL. For ISW3B = 0, the call to the boundary layer routine BLAYER is suppressed in BLCALL.

The step size in the nonequilibrium integration is often quite small, especially in the region upstream of the throat, where some of the reactions may still be nearly in equilibrium. To avoid the excessive output which would result if the flow conditions were printed out for every step in such regions of small step size, NATA prints the results of the nonequilibrium solution at intervals of TPRINT in the nondimensional temperature CT. The control parameter TPRINT is preset to 0.01, and is under input control (with the input name TPRNTI). A counter (NSTEPS) is used to determine the number of steps computed between successive printouts of the flow conditions. This information is included in the printed output.

If the flow solution is being generated by the perturbation technique (INEQ = 0), the temperature perturbation PCT is added to CT before subroutine GUT2 is called to produce the output. Then PCT is subtracted to restore CT to its equilibrium flow value.

3.42 Subroutine PUTQIN

The computation of the cross sections $\bar{\Omega}_{ij}^{(f,s)}$ required in the transport property calculations for any particular conditions of gas temperature, pressure, and composition is carried out in the NATA code under the general control of subroutine PUTQIN. The subroutine selects data for each step of the computations from the edited cross section data in common block /TRANS7/, calls the appropriate subroutine to carry out the computations, and then returns the computed
cross sections for all species pairs to subroutine TRANSP in the \( Q(K,I,J) \) array of common block /TRANS1/ (see the discussion of subroutine TRANSP). The argument \( X \) of subroutine PUTQIN is the partial pressure of electrons in the specified gas mixture, in atmospheres, a quantity which is needed in the computation of effective Coulomb cross sections from equations I(100).

Before beginning the cross section computations, subroutine PUTQIN initializes \( Q(K,I,J) \) to zero for all pairs of species included in the cross section computations.* If the first entry in the \( V \) array is positive, as it should always be for proper operation of the code, \( Q(1,2,1) \) is set to 0.8 \( Q_C \), where \( Q_C \) is the quantity defined by equation I(100b), for later use in the computation of effective Coulomb cross sections for the mixture. The remainder of the subroutine down to statement 240 then consists of a loop which goes through the steps \( L \) of the cross section computation one by one and accumulates the values of the cross sections for all species included in the computations in the array \( Q(K,I,J) \). In this computation, the index \( K = KQ(L) \) indicates the option to be used in performing the \( L \)th step of the computations (see Section 4.6 of Volume II), \( M \) is the location of the first parameter for the \( L \)th step of the computations in the parameter array \( V \), and \( LQ1 \) and \( LQ2 \) are the locations respectively of the first and last pairs in the \( IQ, JQ \) array to which the computations of the \( L \)th step are to be applied. For each step \( L \) of the computations, subroutine PUTQIN determines the option which is to be used for that step from the \( KQ \) array and then calls the appropriate subroutine to carry out the computations for that option, as indicated in Table II. For the Coulomb cross sections \( (KQ = 2) \) however, these computations are carried out in subroutine PUTQIN itself using the previously computed value 0.8\( Q_C \).

*Note that the number \( N \) of species in the cross section computations, which is stored in common block /TRANS8/, is not necessarily the same as the number of species in the transport calculations which is stored in common block /TRANS5/.
As indicated in Table II, the subroutines called by PUTQIN may be divided into three general types, based on the manner in which they return their output. The first and most commonly used type 1, returns the three cross section values $\Omega(1,1), \Omega(2,2)$ and $B^* \Omega(1,1)$ computed for the step to subroutine PUTQIN in the three locations $\Omega(M(1), \Omega(M(2),$ and $\Omega(M(3)$ respectively of the dimensioned variable $\Omega.$ The loop from statement 190 through 200 of PUTQIN then adds these computed values to the previously computed cross sections $Q(K, IQ, JQ),$ for each pair of species in the IQ, JQ array to which the step applies. For the type 2 subroutines, on the other hand, the cross sections are loaded directly into the Q array by the subroutine which calculates them, and statements 190 through 200 in PUTQIN are bypassed. The reason for this difference is that for the type 2 subroutines the computed values of the cross sections for a species depend on the indices of the species, so that a single set of cross section values cannot be returned for all pairs of species to which the step applies.
The single type 3 subroutine in the code, QEX (charge exchange cross section), returns the computed cross section values for the step in the variable $\Omega$ as do the type 1 subroutines. However, in this case, the coding of subroutine PUTQIN offers a choice, determined by the input values of $V$ for the option, $s$ to whether the computed cross sections for the step will be added to the previous cross sections by transferring control to statement 190, as for a regular type 1 subroutine, or will be used to replace the previous values of $\Omega(1,1)$ and $B^* \Omega(1,1)$ by transferring control to statement 170 and bypassing 190.

The final section of coding between statements 210 and 240 of PUTQIN produces a voluminous dump of cross section data under control of the input parameter ISW8B. For ISW8B $\leq 0$, this dump is not produced. For ISW8B $> 0$, the dump is produced the first time PUTQIN is called and every ISW8Bth time thereafter. Note that PUTQIN is called once each time subroutine TRANSP is executed to compute transport properties.

For each step in the cross section calculations, the dump includes a line of output giving the number of the step; the code number KQ of the cross section option used in the step; the positions in the $\gamma$, JQ arrays of the first (LQ1) and the last (NQ(L)) pairs of species for which cross sections are calculated in the step; and the position M of the first unused parameter in the V array. Below this line of indexing information, the calculated values of the averaged cross sections $\Omega_{ij}$ for all pairs of species in the mixture are printed out as they exist at the conclusion of the step. Together with the initial edit of the cross section calculations (Section 3.9 of Volume II), this PUTQIN dump allows detailed checking of the operation of the cross section calculations for a new gas model, since the effect of each step in the calculations upon each pair cross section $\Omega_{ij}$ is clearly shown. The routine could easily be modified to print $\Omega_{ij}$ and $B^* \Omega_{ij}$, in addition.

3.43 Subroutine QCFL

Subroutine QCFL computes the factor $0.8Q_c$ in the Coulomb cross section from equation I(100b). Inputs for this computation are the temperature $T$ in $^\circ$K and the
electron pressure \( X = n_e kT \) in atmospheres. The computed value of \( 0.80 \) in \( 10^{-16} \) cm\(^2\) is returned in \( \Phi M_1 \). For \( C < 0 \), \( \Phi M_1 \) in I(100b) is set to 1.0 while for \( C > 0 \) it is calculated correctly from I(100c) and I(100d). To prevent overflow when \( n_e = 0 \), a small quantity (\( \approx 10^{-50} \)) is added to \( X \) before carrying out the computations.

### 3.44 Subroutine QEX

Subroutine QEX computes the averaged cross sections \( \Omega \) for a pair of species between which there is an exchange interaction (i.e., an interaction such as charge exchange in which the identities of the two species are interchanged) having a cross section of the form

\[
Q_{\text{ex}} = (A-B \log_{10} v)^2
\]

where \( v \) is the relative velocity in cm/sec. The formula for the averaged cross sections \( \Omega \) is then (ref. 9):

\[
\Omega(\ell, s) = 2 \left[ A - \alpha_s B - \frac{1}{2} B \ln(T/W) \right]^2 + 2 \beta_s B^2
\]

(127a)

where

\[
\alpha_s = \frac{1}{2} \left[ \ln(4k/N_0) + \sum_{n=1}^{s+1} \left( \frac{1}{n} \right) - \gamma \right] \log_{10} e
\]

(127b)

\[
\beta_s = \frac{1}{4} \left[ \frac{n^2}{6} - \sum_{n=1}^{s+1} \left( \frac{1}{n^2} \right) \right] \left( \log_{10} e \right)^2
\]

(127c)

\( \gamma = 0.577216 \ldots \) is Euler's constant, and \( W \) is the molecular weight. Putting the numerical values into equations (127) and I(91) then gives the formulas used for \( \Omega \) in the subroutine. Since the cross section \( \Omega \) is not affected by exchange (ref. 10), subroutine QEX sets the contribution \( \Phi M(2) \) to this cross section equal to zero.
3.45 **Subroutine QEXP**

Subroutine QEXP computes the averaged collision cross sections $\bar{\sigma}^{(1,1)}$, $\bar{\sigma}^{(2,2)}$, and $B^* \bar{\sigma}^{(1,1)}$ in $10^{-16}$ cm$^2$ for a pair of species interacting according to the exponential potential $\phi = Ae^{-r}/\rho$, where $r$ is the distance between the two species in Å. These cross sections are obtained by linear interpolation in Monchick's tables (ref. 11) of $\alpha$ versus $I^{(1,1)}$, $A^*$, and $B^*$, where

$$\alpha = \ln (A/kT)$$

$$I^{(1,1)} = \frac{\bar{\sigma}^{(1,1)}}{(4\pi \alpha^2 \rho^2)}$$

and

$$A^* = \frac{\bar{\sigma}^{(2,2)}}{\bar{\sigma}^{(1,1)}}$$

which are stored in common block TRANS4. The table lookup and interpolation are carried out by a call to subroutine QINTRP. The coding assumes that there are 50 entries in these tables.

3.46 **Subroutine QINTRP**

Subroutine QINTRP is used by several of the other cross section subroutines for interpolating tabular data stored in common block TRANS4. In this interpolation, the independent variable is assumed to be stored in monotonically increasing order in the array TL, starting at location N1, and the corresponding values of the three dependent variables in the corresponding locations of the $\Omega$MEGAL, ASTAR and BSTAR arrays. The argument N of the subroutine represents the number of data points to be included in the table, and $A$ and $B$ are respectively the values of the independent and dependent variables for the interpolation. The subroutine searches the data in the TL array for the specified value $A$ of the independent variable and, if $A$ is found to lie within the range of the table, carries out a linear interpolation of the tabular data to determine the values $B(J)$ of the three dependent variables corresponding to the specified value $A$ of the independent variable. If, on the other
hand, A lies outside the range of the table, the last two data points in the table are linearly extrapolated to determine the values of the dependent variable B and, if ISW8B ≠ 0, a message is printed out indicating that an extrapolation of the data was required. This message gives the location in the TL array at which the extrapolation occurred, the desired value of the independent variable, and the two values of the variable from the TL array used in the extrapolation.

3.47 Subroutine QLOJ

Subroutine QLOJ determines the collision cross sections \( \bar{\Omega}(1,1), \bar{\Omega}(2,2) \) and \( B^* \bar{\Omega}(1,1) \) at temperature \( T \) for a pair of species interacting according to the Lennard–Jones 6-12 potential \( \phi = 4\varepsilon \left[ (\sigma/r)^{12} - (\sigma/r)^6 \right] \) by linear interpolation in tables of \( T^* = kT/\varepsilon \) versus \( \Omega(1,1)^* = \Omega(1,1)/\pi \sigma^2 \), \( A^* = \Omega(2,2)/\Omega(1,1)^* \), and \( B^* \) which have been computed for this potential by Monchick and Mason (ref. 12). These tables contain data for 37 reduced temperatures \( T^* \) from 0.1 to 100 and are stored in the arrays TL, QMEGAL, ASTAR, and BSTAR respectively in common block/TRANS4/, starting at location NL = 501 in the arrays.

3.48 Subroutine QMXI

Subroutine QMXI computes cross sections from the empirical mixing rule \( I(102) \) and adds them to the previously computed cross sections \( Q(K,I,J) \) for a series of species pairs given in the IQ, JQ arrays in common block/TRANS7/. The arguments LQ1 and LQ2 of the subroutine indicate the positions of the first and last pairs in the IQ, JQ array to which these computations are to be applied.

To avoid computing the square roots required in equation \( I(102) \) several times in a single call to QMXI, these quantities are stored in the array

\[
SQT(K,I) = \frac{1}{2} \sqrt{Q(K,I,I)} \quad (129)
\]
Upon each entrance to the subroutine, the values of $SQT(I,I)$ are set to 0.0 for all species $I$, to indicate that they have not yet been calculated in the current call to QMIX. (Note that values of $SQT$ cannot be saved over from one call to QMIX to the next, since the values of the cross sections $Q(K,I,I)$ may have changed between the two calls.) The remainder of the subroutine then consists of a loop over the species pairs $I$, $J$ which are to be included in the computation. For each pair, the subroutine checks the values of $SQT(I,I)$ and $SQT(J,J)$ to determine if they have been calculated previously in the loop, and, if not, computes the required values of $SQT$ from (129) for $K = 1, 2, 3$. The $SQT$ values for the pair are then used in equation (102) and the results added to the previously computed cross sections $Q(K,I,J)$ to obtain the final cross section values for the pair.

Since the edited cross section data prepared by subroutine XSECT may contain steps using the option $KQ = 10$ which do not apply to any species pairs, a provision to bypass the cross section computations completely when $LQ1 > LQ2$ is included in the subroutine QMIX to allow for this possibility.

3.49 Subroutine QREPP

Subroutine QREPP computes the averaged cross sections $\bar{\Omega} (1,1)$, $\bar{\Omega} (2,2)$, and $B*\bar{\Omega} (1,1)$ in units of $10^{-16}$ cm$^2$ for a pair of species which interact according to an attractive or repulsive inverse power potential of the form $\phi = Ar^{-\gamma}$, where $r$ is the distance between the species in $\hat{R}$. The cross section formulas for this potential are (ref. 13):

$$\bar{\Omega} (1,1) = \varnothing_{Mega}(ITL) / (\eta^{2/\eta})$$

$$\bar{\Omega} (2,2) = \text{ASTAR}(ITL) * \bar{\Omega} (1,1)$$

$$B*\bar{\Omega} (1,1) = \text{BSTAR}(ITL) * \bar{\Omega} (1,1)$$

(130)

where $\varnothing_{Mega}(ITL)$, ASTAR(ITL), and BSTAR(ITL) are input quantities given by equations I(105).
3.50 Subroutine QSAME

Subroutine QSAME sets the cross sections for a species pair equal to a constant multiple of those computed previously for another pair \((I,J)\). The coding of the subroutine permits a different constant factor to be used for each of the three cross sections \(\tilde{\Omega}(1,1)\), \(\tilde{\Omega}(2,2)\), and \(B^* \tilde{\Omega}(1,1)\).

3.51 Subroutine QTAB

Subroutine QTAB obtains the values of the cross sections \(\tilde{\Omega}(1,1)\), \(\tilde{\Omega}(2,2)\), and \(B^* \tilde{\Omega}(1,1)\) for a pair of species at a given temperature \(T\) by linear interpolation in tables of \(\tilde{\Omega}(1,1)\), \(\tilde{\Omega}(2,2)\) and \(B^* \tilde{\Omega}\) versus temperature for the species which are included in the input. The tabulated data are assumed to be given in the arrays MEGA1, ASTAR, BSTAR, and TL, respectively, in common block TRANS4, with the tables beginning at location NL in the arrays and containing NL data points. An option is also provided to multiply the tabulated cross sections by an arbitrary constant factor \(A\) in the computations.

3.52 Subroutine Q11

Subroutine Q11 multiplies the previously computed cross sections for a series of species pairs by the temperature dependent factor \(f(T)\) given by equation I(108). The subroutine computes the value of this factor at the specified temperature \(T\) from the input data for the subroutine and then calls subroutine Q14 with the appropriate inputs to carry out the actual multiplication of the cross sections by this factor.

3.53 Subroutine Q12

Subroutine Q12 computes the cross sections \(\tilde{\Omega}(1,1)\), \(\tilde{\Omega}(2,2)\), and \(B^* \tilde{\Omega}(1,1)\) for a species pair from the generalized mixing rule given by equation I(109). As with the other type 1 subroutines discussed above, it returns the computed cross sections to subroutine PUTQIN in the dimensioned variable \(\phi M\).
3.54 **Subroutine Q13**

Subroutine Q13 computes the cross sections $Q(K_l, I, J)$ according to equations I(110) and I(111) for a fixed value of $K_l$ specified in the cross section data ($1 \leq K_l \leq 3$) and for a series of species pairs $I, J$ given in positions LQ1 through LQ2 of the IQ, JQ arrays. The computed cross sections then replace the previous values for these cross sections in the $Q(K, I, J)$ array.

3.55 **Subroutine Q14**

Subroutine Q14 computes the cross sections $\tilde{\Omega}(1, 1)$, $\tilde{\Omega}^2 (2, 2)$, and $B \times \tilde{\Omega} (1, 1)$ for a series of species pairs by multiplying the previously computed values of these cross sections by a constant factor, as indicated in equation I(112). The computed cross sections then replace the previous values for these cross sections in the $Q(K, I, J)$ array.

3.56 **Subroutine RADIUS**

This small routine performs geometric calculations that are required at several points in the computation of the boundary layer at the first flow point in subroutine BLAYER. The inputs to RADIUS are the type of nozzle geometry ITYPE, the axial coordinate $X$, and the profile index $L$. The quantities computed and returned to BLAYER are the profile ordinate $R$, the geometric area ratio $AG$, and the factor $AGJ$, which is proportional to $r^{2j}$ in equation I(171).

For ITYPE = 1 (two-dimensional nozzle), $R$ is obtained by calling the entry GMAR of subroutine GEOMAR. Then $AG = K/R_0$ where $R_0$ is the half-width of the throat gap, and $AGJ = 1$ since $j = 0$.

For ITYPE = 2 (axisymmetric nozzle), $R$ is again obtained by calling GMAR, $AG = (R/R_0)^2$, where $R_0$ here is the throat radius, and $AGJ = AG$ since $j = 1$.

For ITYPE = 3 (rectangular channel), the ordinates $YZ(1), YZ'(2)$ of the two profiles are obtained by calling the entry GMAR2 of subroutine GEOMAR, $AG = YZ(1) \cdot YZ(2)/Y_0Z_0$ where $Y_0Z_0$ is the product of the profile ordinates at the throat, $R = YZ(L)$, and $AGJ = [YZ(M)]^2$, where $M = 2$ if $L = 1$ and $M = 1$ if $L = 2$. 

-160-
3.57 Subroutine READ

READ is the input routine for NATA. It reads the input data, sets up the geometric description and gas model for the problem, and initializes most of the control parameters.

3.57.1 Input Operations

The subroutine begins by reading the input data for the current case. The routine contains four READ statements, of which the first two are always executed. The first of these statements reads a comment card into the array ACOM. This information is subsequently written out as part of a heading for the output for the case. The second READ statement reads data for the variables included in namelist INPUT. This is the main input operation for the code. Namelist INPUT contains all of the inputs to NATA except those required for defining chemical elements, species, and reactions (read under namelist EINPUT), and those required for calculating the transport cross sections of species (read under namelist TINPUT). When the code is run using the chemical and cross section data compiled into the BLOCK DATA routines, as it is normally, the statements READ(5, EINPUT) and READ(5, TINPUT) are not executed.

All of the variables in INPUT are printed out at the head of the output for the case by the statement WRITE (6,INPUT). If EINPUT and TINPUT are read, their variables are also printed out in namelist format. The reason for breaking the code inputs into these three groups is to minimize the number of pages of such output in normal usage, when EINPUT and TINPUT are not used.

3.57.2 Sonic Mass Flux Calculation

If ISW2B ≤ 0, NATA determines the equilibrium conditions in the reservoir from data on the reservoir pressure or stagnation enthalpy and the total mass flow. In these cases, the sonic mass flux SMASS is required in the calculations to determine the reservoir temperature. This quantity is computed in READ from the total mass flow Ê (denoted by FLW) and the geometric dimensions of the throat. The coding treats three cases:
(1) Two-dimensional nozzle

Here \( \dot{M} \) and \( \text{FLOW} \) denote the mass flow per unit length of the throat gap. Hence

\[
\text{SMASS} = \frac{\dot{M}}{D}
\]

(131)

where \( D \) is the throat gap. Since \( \text{SMASS} \) is required in \( \text{gm/cm}^2\text{-sec} \) while \( \dot{M} \) is in \( \text{lb/in}^2\text{-sec} \) and \( D \) in inches, a numerical conversion factor from \( \text{lb/in}^2\text{-sec} \) to \( \text{gm/cm}^2\text{-sec} \) is inserted on the right. This factor is \( 453.5924/(2.54)^2 = 70.3069 \).

(2) Axisymmetric nozzle

In this case, \( \dot{M} \) is the total mass flow in \( \text{lb/sec} \), and

\[
\text{SMASS} = \frac{4\dot{M}}{\pi D^2}
\]

(132)

Insertion of the conversion factor \( 70.3069 \) gives a numerical coefficient of \( 4 \times 70.3069/\pi = 89.5173 \).

(3) Rectangular channel

The sonic mass flux here is

\[
\text{SMASS} = \frac{\dot{M}}{D_1 D_2}
\]

(133)

where \( D_1 \) and \( D_2 \) are the throat gaps of the channel's two profiles.

3.57.3 Automatic Air Model Selection

If the input AAMS (acronym for "automatic air model selection") is .TRUE. and IGAS is either 1 or 2, subroutine READ resets IGAS to either 1 or 2 on the basis of an enthalpy or temperature criterion. IGAS = 1 gives the large air model, including five ion species, which is required in cases
with high reservoir temperatures. IGAS = 2 gives the small air model which includes only a single ion (NO\(^+\)) and which can be used at lower temperatures. If the reservoir temperature is read in (ISW2B > 0), the air model is selected to be IGAS = 1 or 2 according as the reservoir temperature CTAPI is greater than or less than 6000\(^\circ\)K. On the other hand, if the reservoir temperature is to be determined from mass flow data (ISW2B \(\leq 0\)), then its value is not yet known at the time when READ is executed. In this case, the air model selection is based on a stagnation enthalpy criterion of 8000 Btu/lb. For ISW2B < 0, the stagnation enthalpy is an input. For ISW2B = 0, the inputs are stagnation pressure and mass flow; in this case, the stagnation enthalpy is estimated from the Winovich correlation (ref. 14). The Winovich estimate of the stagnation enthalpy is computed from the formula

\[
\text{HWINOV} = \left[ \frac{136.7 \; p_0}{m_0} \right]^{2.52}
\]

(134)

where \(p_0\) denotes the reservoir pressure in atmospheres and \(m_0\) the sonic mass flux in \(g/cm^2\;sec\). The numerical factor on the right in (134) differs from the coefficient (280) in Winovich's equation because it includes a factor converting \(m_0\) from \(g/cm^2\;sec\) to \(lb/ft^2\;sec\).

3.57.4 Initializations Based on Inputs

With few exceptions, the input variables of NATA are not altered by execution of the program. Also, the Namelist input system resets only those variables which are actually referenced in the input cards for a given case. Thus, when several cases are run in a single job, the input cards for the second and subsequent cases need contain data only for the variables which the user wishes to change, e.g., the pressure, mass flow, nozzle index, etc.

The input-determined variables which are not altered by program execution are included directly in the namelists INPUT, EINPUT, and TINPUT. However, some input-determined variables are altered by the program. In such cases, the variable which is altered and its corresponding input are given different names and thus are stored in separate
locations. The input name is the internal variable name (or a contraction thereof) with the letter I added at the end. At the cost of some increase in storage requirements, this arrangement protects the input variables against alteration during execution.

3.57.5 List of Test Sections

The data in the input array TSDIAM specify points in the flow at which special calculations are to be done. In the case of a two-dimensional or axisymmetric nozzle, the entries in TSDIAM are nozzle diameters at which model-condition calculations are to be performed. For a channel, the TSDIAM entries are specified channel widths at which free-stream flow and boundary layer calculations are to be performed.

Subroutine READ uses the data in the TSDIAM array to set up another array, TSAR, whose values are actually used by the program to select the points at which the special calculations are done. For a nozzle, the TSAR entries are geometric area ratio values. For a channel, the TSAR are widths in centimeters. In generating the TSAR array, READ rejects any TSDIAM values which are less than or equal to the throat diameter for the profile involved, and determines the number (NTS) of valid TSAR entries. All the entries of TSDIAM are preset (in BL@CK DATA) to the value $1 \times 10^{20}$. Thus, any values in excess of $1 \times 10^{20}$ are assumed not to have been set by the user in input, and are not regarded as valid points.

3.57.6 Geometric Description

A nozzle is specified by one profile curvefit, a channel by two. For each profile, the parameters defining the geometry of the nozzle or channel may be either read in as part of the input or, if one of the standard geometries is desired, obtained from pre-coded data. For the Jth profile ($J = 1$ or 2), if NPR@FL(J) is nonzero, the geometric data are looked up in the ZPRP array, which is set in BL@CK DATA. This array occupies the common block /N@ZZ/. If NPR@FL(J) is zero, the geometric data are set on the basis of the geometric input variables such as NSECTS, DIAM, and DARAMI.

-164-
In either case, after the geometric model has been set up, subroutine READ prints out a summary description of it, unless the standard geometry specified is the same as in the preceding case in the job.

3.57.7 Extra Chemical Elements

If additional chemical elements have been defined in the EINPUT input, the data for them are loaded into the EPRP array, which contains the precoded data for elements. The number of input-defined elements is NEELS. For the Ith such element, the atomic number is obtained from EEPRP(1,I) and the atomic weight from EEPRP(2,I). The index assigned to the element in the master list of elements is IEEP(I). The Hollerith atomic symbol for the element is obtained from the array ASYM, which is set in a data statement in READ.

3.57.8 Gas Model

The elements, species, and reactions present in the gas mixture are specified by the group of variables ISC, ISS, ISR, IC, IE(I), NSC, QPJ(I), JCS(I), IS(I), and IR(I). If IGAS=0, these variables are obtained from the input data. If IGAS is positive, they are looked up in the array of precoded data, GPRP(I,IGAS), which occupies common block /MIXT/. The physical definitions of the entries in this array are given in Section 4.5 of Volume II. If IGAS is negative, the input value is saved in the location IGS, and IGAS is reset to its own absolute value. In this case, all of the above-listed variables except QPJ(I) are again looked up in GPRP, but the QPJ (the mole fractions of the cold species) are obtained from input. This provision allows input control of the overall elemental composition of the standard gas mixtures containing more than one chemical element.

After these general specifications of the gas model have been determined, the required properties of the chemical elements, species, and reactions are looked up. The data for elements are obtained from the array EPRP(I,IE), where the second subscript is the index in the master list of elements. The species data are looked up in the array SPRP(I,IS), where IS is the index in the master list of species. The reaction data are obtained from RPRP(I,IR),
where IR is the index in the master list of reactions. The entries in these arrays are all defined in Section 4 of Volume II. The order of the elements, species, and reactions in a gas model is determined by the index arrays IE(I), IS(I), IR(I), and may differ from the order in the corresponding master lists.

3.57.9 Geometric Sequence of Model Points

Positions at which model condition calculations are to be done can be determined by specifying an initial point XMØDPL, a final point CXMAXI, and a number of points NMØDPT. From these data, subroutine READ computes a factor FACMP which is used to generate a geometric sequence of NMØDPT values of x extending from XMØDPL to CXMAXI. This factor is calculated from the formula

$$\text{FACMP} = \left[ \frac{\text{CXMAXI}}{\text{XMØDPL}} \right] \frac{1}{(\text{NMØDPT}-1)}$$

(135)

3.58 Subroutine RESET

This Fortran routine, used only in the IBM version of NATA, simulates the UNIVAC 1108 timing routines RESET and TIME. The time data are obtained by calling the library routine ACUCPU, which returns in its argument ICPU the number of milliseconds of CPU time remaining before the TIME parameter of the job step is reached. Subroutine RESET is called by the main program near the beginning of each NATA job. The initial value of ICPU is stored in IZERO. Subsequent calls to the entry TIME then give the elapsed time I in milliseconds.

3.59 Subroutine RESTMP

RESTMP performs the iterative solution for the reservoir conditions when the second or third option for input specification of these conditions is used. These options have been described in Section 6.5 of Volume I. Usage of these options is controlled by the indicator ISW2B. For ISW2B = 0, the second option is used, and for ISW2B < 0, the third option. (For ISW2B > 0, the first option is used and RESTMP is not called.)
3.59.1 Second Option

The second option is implemented by the coding down to statement 90. In this option, the input specification of the reservoir conditions is based on the reservoir pressure, \( p_0 \), and the total mass flow, \( \dot{M} \). In the code, the value of \( p_0 \) in atmospheres is available in the location PRESA in unlabelled common, and the sonic mass flux \( m_\ast \) in \( g/cm^2\text{sec} \) is given by SMASH in common block /MASSFL/.

The basic problem in the second option is to determine the reservoir temperature, \( T_0 \), from these data. The code capabilities available to support this effort are

1. Entry INTA of subroutine INGAS, which computes all of the conditions in the reservoir from \( p_0 \) and \( CTAP = T_0 \).

2. Subroutine NRMAX, which computes the sonic mass flux \( m_\ast \) for equilibrium flow, based on given reservoir conditions.

RESTMP determines \( T_0 \) by an iteration in which INTA and NRMAX are called, at each step, and \( T_0 \) is repeatedly adjusted to make the calculated sonic mass flux nearly equal to SMASH.

RESTMP starts the iteration with a standard guess \( T_0 = 10000 \text{K} \). Subroutines INTA and NRMAX are then called to compute the reservoir conditions and sonic mass flux based on the trial value of \( T_0 \). The call to INTA must be preceded by a call to subroutine INIT to set a number of species properties which are nondimensionalized using the reservoir temperature.

The sonic mass flux computed by NRMAX is in the non-dimensional form

\[
SM = \frac{\rho_\ast u_\ast}{\rho_0 u_s} = \frac{m_\ast}{\rho_0 \sqrt{R_0 T_0 / p_0}}
\]

From the perfect gas law,

\[
P_0 = \frac{\rho_0 R_0 T_0}{\mu_0}
\]
Elimination of $\rho_0$ between (136) and (137) gives the relation

$$T_0 = \frac{W_0(T_0)}{R_0} \left[ \frac{P_c \cdot SM(T_0)}{m_*} \right]^2$$

(138)

which is satisfied by the correct value of the reservoir temperature $T_0$. In (138), $W_0$ denotes the reservoir molecular weight (CMA), $R_0$ the universal gas constant in mechanical units $(8.31434 \times 10^7$ erg/mole K), $P_c$ the reservoir pressure in dyne/cm$^2$, $SM(T_0)$ the nondimensional sonic mass flux computed by NRMAX, and $m_*$ the sonic mass flux (SMASS, g/cm$^2$sec) based on the input total mass flow $\dot{M}$. Equation (138) may be rewritten

$$T_0 = \frac{W_0(T_0)}{C} \left[ \frac{SM(T_0)}{P_c} \right]$$

(139)

where the parameter $C$ (denoted by 'CONST' in the program) is a constant of the iteration:

$$C = R_0 \left[ \frac{m_*}{P_c} \right]^2$$

(140)

The second estimate of $T_0$ is obtained using equation (139), with $W_0$ and SM values based on the first trial value. The quantity SM is found to vary only weakly with $T_0$, while the molecular weight is monotonically decreasing function of $T_0$ at constant pressure. Thus, the error in the second estimate of $T_0$ is opposite in sign to that of the first estimate; the two values bracket the correct solution. Accordingly, after INIT, INTA, and NRMAX have been called, the third estimate of $T_0$ is obtained by a linear interpolation technique. The function $F(T_0)$ is approximated as a linear function between $T_0 = T_1$ and $T_0 = T_2$ (the first two estimates). Then

$$\frac{F_2 - F_2}{T_0 - T_2} = \frac{F_2 - F_1}{T_2 - T_1}$$

(141)

Elimination of $F$ between (139) and (141) gives
for the third estimate.

Beginning with the fourth estimate, the dependence of \( F(T_0) \) upon \( T_0 \) is approximated by a quadratic function based upon the data from the three most recent iterations:

\[
F = D_1 + D_2 T_0 + D_3 T_0^2
\]  

(143)

Substitution of the data from the three points gives a system of linear equations for \( D_1, D_2, D_3 \). This system can be written in matrix form.

\[
A \cdot D = B
\]  

(144)

where

\[
A = \begin{bmatrix}
1 & T_1 & T_1^2 \\
1 & T_2 & T_2^2 \\
1 & T_3 & T_3^2
\end{bmatrix}
\]  

(145a)

\[
B = \begin{bmatrix}
F_1 \\
F_2 \\
F_3
\end{bmatrix}, \quad D = \begin{bmatrix}
D_1 \\
D_2 \\
D_3
\end{bmatrix}
\]  

(145b)

This system of equations is solved using subroutine SIMQ, which places the solution vector \( D \) into the locations originally occupied by the vector of constants \( B \). Then elimination of \( F \) between (139) and (143) gives

\[
T_0 = C \left( D_1 + D_2 T_0 + D_3 T_0^2 \right)
\]  

(146)

The solution of this quadratic equation for \( T_0 \) is

\[
T_0 = -\frac{T_2 F_1 - T_1 F_2}{T_2 - T_1} \cdot \frac{1}{\frac{F_2 - F_1}{T_2 - T_1} - C}
\]  

(142)
The sign before the radical is determined by noting that, for $D_3 \to 0$, the solution must approach $3$.

$$T_0 = \frac{1}{2D_3} \left[ - (D_2 - \frac{1}{C}) \pm \sqrt{(D_2 - \frac{1}{C})^2 - 4 (D_1 D_3)} \right]$$  \hspace{1cm} (147)

Thus, the sign represented by $+$ must be opposite to that of $-(D_2 - 1/C)$, i.e., the same as that of $(D_2 - 1/C)$.

In the program, the temperature values $T_1$, $T_2$, $T_3$ are stored in the array CTSAVE and the parameter values $F_1, F_2, F_3$ in the array F. If SIMQ returns an error indicator $KS \neq 0$, which will occur if the matrix (145a) is singular, RESTMP reverts to the linear interpolation technique.

The convergence criterion for the iteration is that the relative change in $T_0$ between two successive iterative steps be smaller than or equal to $10^{-4}$.

3.59.2 Third Option

The coding of the third option begins with statement 90 of RESTMP and continues to the end of the subroutine.

In this option, the reservoir conditions are specified by input of the total mass flow $\dot{M}$ and the stagnation enthalpy $h_0$. The latter quantity is given, in calories per gram, by the variable HS in common /RDMAIN/. The third option requires a double iteration to determine both the reservoir temperature $T_0$ and the reservoir pressure $p_0$ from the available data.

To start the iteration, an initial estimate of the reservoir pressure is obtained from an approximate sonic flow analysis for a perfect gas. For a gas with constant specific heats,

$$\rho^* = \left( \frac{2}{\gamma + 1} \right)^{1/\gamma - 1} \rho_0$$  \hspace{1cm} (149a)
These relations are derived, for example, in Liepmann and Roshko (ref. 15, pp. 51-54). Combination of equations (149) and (137) gives

\[ u_* = a_* = \sqrt{\frac{2}{\gamma + 1}} \quad a_0 \quad (149b) \]

\[ a_0 = \sqrt{\frac{\gamma_p}{\rho}} = \sqrt{\frac{\gamma R_0 p_0}{T_0}} = \sqrt{(\gamma - 1)} h_0 \quad (149c) \]

In RESTMP, an initial estimate of \( p_0 \) is obtained from equation (150) using a standard estimate of the specific heat ratio, \( \gamma = 4/3 \). This value gives

\[ p_0 = \frac{\gamma - 1}{\gamma} \sqrt{\frac{\gamma + 1}{2(\gamma - 1)}} \left( \frac{\gamma + 1}{2} \right)^{1/(\gamma - 1)} \quad m_* \sqrt{h_0} \quad (150) \]

in which all quantities are assumed to be expressed in absolute cgs units. With \( p_0 \) in atmospheres and \( h_0 \) in cal/gm, this relation becomes

\[ p_0(\text{atm}) = 4.8 \times 10^{-3} \quad m_* \sqrt{h_0} \quad (\text{cal/g}) \quad (152) \]

The iteration scheme used to solve for the reservoir temperature and pressure is illustrated schematically in figure 25. The starting point for the iteration (labelled "1") lies at the pressure (152) and an initial estimate of temperature, 10,000°K. In the first stage of the double iteration, the temperature is varied at constant pressure to make the reservoir enthalpy equal to the input value \( H_S \) (denoted by \( h_S \) in the figure). The coding of this stage runs from statement 110 down to (but not including) 130. At each step of this iteration on temperature, INIT and INTA are called, and the reservoir enthalpy is computed from CHA using equation I(241). In the first step, the temperature
Figure 25. Iteration Scheme of Third Option for Specifying Reservoir Conditions
estimate is improved by assuming that the enthalpy is approximately proportional to the temperature:

\[ T_0 = T_1 \cdot \frac{h_s}{h_1} \tag{153} \]

where \( T_1 \) is the initial estimate of \( T_0 \) and \( h_1 \) is the specific enthalpy corresponding to \( T_1 \). In subsequent steps, the relation between \( T \) and \( H \) is approximated by a linear relation which does not necessarily pass through the origin:

\[ T_0 = T + (h_s - h) \left( \frac{T_p - T}{h_p - h} \right) \tag{154} \]

This iteration on temperature is continued until the calculated enthalpy is equal to the desired value to within 1 part in \( 10^7 \). The result of this stage is the point "2" in figure 25.

In the second stage, the reservoir pressure is varied at constant reservoir temperature to make the sonic mass flux equal to its desired value, SMASS. This stage is implemented by the coding from statement 130 down to (but not including) 150. In each cycle of this stage of the iteration, subroutine NRMAX is called to compute the non-dimensional sonic mass flux, SM. The sonic mass flux \( m^* \) in g/cm²sec is calculated from this value using equation (136). The estimate of the reservoir pressure is then improved on the assumption that \( m^* \) is directly proportional to \( p_0 \), and subroutine INTA is called to determine the reservoir conditions based on the new \( p_0 \) value. This stage is continued until the calculated sonic mass flux agrees with the desired value to within 1 part in \( 10^5 \). The resulting point is "3" in figure 25.

The complete iteration consists of alternate temperature-adjusting and pressure-adjusting stages, and produces a series of points "4", "5", "6", etc., which spiral in towards the desired solution point in \( T_0 - p_0 \) space (figure 25). After each pair of stages (2,3), (4,5), etc., tests on temperature and pressure are applied to assess the convergence of the entire procedure. The criteria for convergence are that the relative changes in \( T_0 \) and \( p_0 \) during a stage-pair be less than or equal to \( 10^{-4} \).
If the input control parameter ISWA is nonzero, a dump is written in each cycle of this iteration to allow tracing of the subroutine's operation.

3.60 Subroutine RNKT

RNKT is the routine that computes the changes in the dependent flow variables over an integration step using the Treanor-Runge-Kutta scheme explained in Section 7.5.1 of Volume I. Figure 26 is a flowchart of RNKT.

In the initializations, IFAIL is an indicator for the cause of step failure, used in diagnostic output which is produced when ISW5B is set to a negative value in the input (see glossary of Fortran symbols for common block /TNERK/ in Section 4). TE is a logical indicator. A .TRUE. value means that a chemical nonequilibrium model is in use. If TE = .FALSE., an electronic equilibrium model is being used. LIM denotes the number of dependent variables included in the numerical integration. For a chemical nonequilibrium model, LIM = n + 1 and the variables are \( \gamma_j \) for \( j = 1 \) to \( n \) and \( CT = T/T_0 \). For an electronic nonequilibrium model, LIM = n + 3 and the variables are \( \gamma_j, CT, CTE, E-E/T_0 \), and \( \text{CHA} = h_0 W_0/R_0 T_0 \). The energy transfer to the electron gas (QDPE) at the start of the integration step is denoted by QDPEB. Finally, DQMRK is the maximum allowable change in QDPE from QDPEB during the integration step.

For ISW5B different from zero, diagnostic dumps are produced at five locations in RNKT. The coding used to generate these dumps is omitted from the flowchart, as it is not essential to the function of the subroutine.

The computations in RNKT closely follow the analysis given in Section 7.5.1 of Volume I. The values \( y_1, y_2, y_3, y_4 \) are represented by arrays GJ1(J), GJ2(J), GJ3(J), GJ4(J), where \( J \) runs from 1 to LIM. Thus, these arrays contain not only the species concentrations but also the other dependent variables, CT, etc. The derivatives \( f_1, f_2, f_3 \) are represented by arrays F1(J), F2(J), F3(J). The parameter \(-P \Delta x\) in I(398) is represented by an array P(J). The quantities \( F_1, F_2, F_3 \) defined by equations I(398) are represented by X1(J), X2(J), X3(J).
Figure 26. Flowchart of Subroutine BLOCK

ORIGINAL PAGE IS OF POOR QUALITY
Following the initializations, RNKT loads data into the F1, GJ1, and GJ2 arrays. If any of the GJ2 values is negative, FAILED is set to .TRUE. and the RETURN is executed. If not, CX is set equal to CXB + HDELX, where HDELX is one-half of the step size DELTAX. Then DERIVS is called to compute the derivatives F2(J), and the GJ3 array is set. The signs of all the GJ3(J) values are tested. In addition, if TE is .FALSE., the new value of the energy transfer QDPE to the electron gas, produced by the call to DERIVS, is tested to determine whether |QDPE - QDPEB| is less than or equal to DQMRK. If not, the error return is executed unless DQMAX = 0 (which means that the current step is the first step in the numerical integration).

If the solution thus far passes these validity checks, DERIVS is called again to determine the derivatives F3(J). Next, the parameters P(J) are calculated. The basic formula for P(J) is I(402). In the Fortran notation, with P(J) — PΔx in I(402), this equation becomes

\[ P(J) = 2 \cdot \frac{F3(J) - F2(J)}{F2(J) - F1(J)} \]  

(155)

However, under certain conditions, P(J) is set to zero. As explained in Section 7.5.1 of Volume I, this is done if (f2 - f1)/f1 = X4(J)/F1(J) has a magnitude smaller than 10^-4, to avoid excessive loss of accuracy on subtractions. P(J) is also set to zero if F1(J) = 0, to avoid a divide check error in the test just described, or if the evaluation of (155) would lead to a floating point overflow on the indicated division. For P(J) ≥ 0, the changes of the dependent variables are computed using the Runge-Kutta formula I(392) in place of Treanor's formula I(401).

Once the P(J) have been determined, RNKT proceeds to compute X1(J), X2(J), and X3(J). If P(J) ≤ -1.25, these quantities are calculated from equations I(398). Since the Fortran P(J) corresponds to (-P Δx) in equations I(398), these formulas can be rewritten in the form
\[ F_1 = X1(J) = \frac{e^{P(J)} - 1}{P(J)} \]  \hspace{1cm} (156a) 

\[ F_2 = X2(J) = \frac{X1(J) - 1}{P(J)} \]  \hspace{1cm} (156b) 

\[ F_3 = X3(J) = \frac{X2(J) - \frac{1}{2}}{P(J)} \]  \hspace{1cm} (156c) 

which is used in RNKT. If \( P(J) > -1.25 \) (but negative), the subtractions in (156) could cause excessive loss of accuracy. In this case, \( F_1, F_2, \) and \( F_3 \) are evaluated from the power-series expansion

\[ F_n = \sum_{k=0}^{\infty} \frac{[P(J)]^k}{(n+k)!} \]  \hspace{1cm} (157) 

which can easily be derived by substituting the Maclaurin expansion for \( e^x \) into (156). In RNKT, the sums (157) for \( F_1, F_2, \) and \( F_3 \) are accumulated in the locations \( S_1, S_2, S_3 \). The general terms in the three expansions are denoted by \( T_1, T_2, \) and \( T_3 \), respectively. These terms are initialized to their values for \( k = 0 \), namely \( T_1 = 1, T_2 = 1/2, T_3 = 1/6 \). Then the successive terms in the series are generated recursively:

\[ (T_n)_k = (T_n)_{k-1} \cdot \frac{P(J)}{n+k} \]  \hspace{1cm} (158) 

where \( (T_n)_k \) denotes the \( k^{th} \) term in the expansion for \( F_n \). In GN, the denominator \( (n+k) \) in (158) is represented by \( (n+Z-3) \), where \( Z \) is initialized to 3 for \( k = 0 \) and is incremented by 1 each time around the iterative loop. The
evaluation of (157) proceeds until the addition of T3 to S3 no longer affects the value of S3 to within floating-point accuracy.

Next, the GJ4(J) array is set. For each J, if P(J) is negative, GJ4(J) is computed from equation I(405). For zero or positive P(J), GJ4(J) is calculated from equation I(394c). After the usual validity checks, the flow variables GJ(J), CT, etc., are set to their corresponding values in GJ4(J), CX is set to CXB + DELTAX, and a final call to DERIVS is executed. The resulting values for the derivatives of the flow variable are loaded into F4(J), and the array SDQ(J) of changes in the dependent variables over the whole integration step is computed. For each J, if P(J) \( \geq 0 \), the Runge-Kutta formula I(392) is used to calculate SDQ(J). If P(J) < 0, then Treanor's formula I(401) is used. Finally, the results are transferred from the SDQ(J) array into the storage locations SDGJ(J), SDT, etc., used by other parts of the program.

3.61 Subroutine SHOCK

Subroutine SHOCK* computes the oblique shock angle and the conditions behind the shock for supersonic flow of a perfect gas over a wedge. The first three of the arguments are inputs to the routine. BMF denotes the free-stream Mach number M, DEL the flow deflection angle (i.e., the angle of attack of the wedge surface, \( \alpha \)), and GAM the specific heat ratio \( \gamma \). The outputs are FL and IERR, which are defined in Section 4.128 and by comment cards in the subroutine listing.

The method of solution is based upon the relation between the shock angle \( \sigma \) and the deflection angle \( \alpha \), which is given, for example, in Liepmann and Roshko (ref. 15, p.87). This relation can be expressed as a cubic equation in \( \sin^2 \sigma \):

\[
A \sin^3 \sigma + B \sin^2 \sigma + C \sin \sigma + D = 0
\]

*This subroutine was programmed by Harvey Buss, Avco Systems Division. Statements for calculating quantities not required in NATA have been converted into "comments" to save storage.
\[ \sin^6 \alpha + b \sin^4 \sigma + c \sin^2 \sigma + d = 0 \quad (159) \]

where

\[ b = -\frac{M^2+2}{M^2} - \gamma \sin^2 \alpha \quad (160a) \]

\[ c = \frac{2M^2+1}{M^4} + \left[ \frac{(\gamma+1)^2}{4} + \frac{\gamma-1}{M^2} \right] \sin^2 \alpha \quad (160b) \]

\[ d = -\frac{\cos^2 \alpha}{M^4} \quad (160c) \]

Equation (159) has three roots for \( \sin^2 \sigma \), in general, of which the lowest corresponds to a decrease in entropy and thus to a physically impossible flow. The lower of the two remaining roots gives the desired weak-shock solution, while the uppermost root gives a strong-shock solution (e.g., a normal shock with \( \sigma = \pi/2 \) for \( \alpha = 0 \)). Subroutine SHOCK first obtains the lowest of the three roots by solving equation (159) using Newton's method. With \( u = \sin^2 \sigma \) and

\[ F(u) = u^3 + bu^2 + cu + d \quad (161a) \]

\[ F'(u) = 3u^2 + 2bu + c \quad (161b) \]

an improved estimate \( u' \) of the root is obtained, at each step of the iteration, using the formula

\[ u' = u - \frac{F(u)}{F'(u)} \quad (162) \]
The convergence criterion is that \( u' = u \) to within 1 part in \( 10^8 \). If convergence is not achieved within 50 iterations, an error message is written and the RETURN is executed.

Use of an initial estimate \( u = 0 \) ensures convergence to the lowest root. Once this root \( u_1 \) has been obtained, the desired weak-shock solution is obtained by factoring the equation \( F(u) = 0 \):

\[
(u-u_1) (u^2 + Au + 3) = 0
\]

Equating the coefficients of corresponding powers of \( u \) in (161a) and (163) gives

\[
A = u_1 + b \quad (164a)
\]
\[
B = \frac{-d}{u_1} \quad (164b)
\]

The two remaining roots of the cubic are then obtained by equating the quadratic factor in (163) to zero:

\[
u = \frac{1}{2} \left[ -A \pm \sqrt{A^2 - 4B} \right] \quad (165)
\]

If the flow-deflection angle \( \alpha \) is too large, the argument of the square root in (165) becomes negative and equation (159) has no physically valid solution. In this case, the error indicator \( IERR \) is set to 0 and the RETURN is executed. When \( A^2 - 4B \) is positive, the lower of the two values (165), \( u_2 \), is equal to \( \sin^2 \alpha \) for the weak shock.

After the shock angle \( \alpha \) has thus been determined, subroutine \text{SHOCK} calculates the conditions behind the shock using perfect gas relations (ref. 15, p. 86). The only result required for use in NATA is the shock pressure ratio,
\[
    \frac{P_2}{P_1} = \frac{2 \gamma M^2 \sin^2 \sigma - (\gamma - 1)}{\gamma + 1} \quad (166)
\]

3.62 **Subroutine SIMQ**

SIMQ is a standard subroutine for solving systems of simultaneous linear equations. It was obtained from the IBM Scientific Subroutine Package (ref. /), and was modified slightly for use in NATA. The modifications consisted of converting it into a double precision routine and the addition of simple coding to allow SIMQ to handle the special case of one equation in one unknown. The only available documentation of SIMQ is that provided by comments in the source program.

3.63 **Subroutine STUNTS**

This routine performs two separate and unrelated operations. When called through the principal entry (STUNTS), it produces tables of species thermal properties as functions of temperature for the species in the specified gas model. When called through the entry STUNT2, it gives an edit of the transport property calculations, and optionally, a table and a punched deck containing averaged species cross section data.

3.63.1 **Thermal Property Edit**

The thermal property edit is produced by the D0 loop ending at statement 110. For each species \( J \), the properties are computed at a preset series of temperatures \( TT(I) \) up to the input value of the reservoir temperature \( CTAP \). These temperatures are as follows:

- From 100\(^\circ\)K to 1000\(^\circ\)K at intervals of 100\(^\circ\)K
- From 1000\(^\circ\)K to 3000\(^\circ\)K at intervals of 200\(^\circ\)
- From 3000\(^\circ\)K to 10000\(^\circ\)K at intervals of 500\(^\circ\)K
- From 10000\(^\circ\)K to 20000\(^\circ\)K at intervals of 1000\(^\circ\)K
- From 20000\(^\circ\)K to 50000\(^\circ\)K at intervals of 2000\(^\circ\)K
- From 50000\(^\circ\)K to 125000\(^\circ\)K at intervals of 5000\(^\circ\)K
The maximum number of temperatures is 74. (However, the thermo fits for the molecular species in the compiled-in models are useful only up to about 20000°K.)

The thermal properties are calculated by calling subroutine THERM at each temperature. THERM is programmed to calculate the properties using two different methods, the physical model and the thermo fit (Section 2.2 of Vol. I, ref. 1). The physical model is used for temperatures more than 500°K below the switchover temperature, CMAXX·CTAP. The thermo fit is used for temperatures more than 500°K above the switchover temperature. In a range 1000°K wide centered at the switchover temperature, a linear combination of the results from the two methods is used, to provide a continuous transition.

To obtain thermal properties based solely on the physical model, rather than on a mixture of the two methods, subroutine STUNTS first sets CMAXX to 1000, and calls THERM for each temperature. With this large value of CMAXX, all of the temperatures are more than 500°K below CMAXX times the reservoir temperature. The results of the thermal property calculations based on the physical model are stored in the array TPR(P(I,K,M) for M = 1. In the case of a nonstandard species for which physical model data have not been provided (ETAJ(J) = 0), the elements of TPR(P for M = 1 are all set to zero and the calculations based on the physical model are skipped.

For each species, if thermo fit data have been provided (IGJ(J) ≠ 0), CMAXX is set to -1000 and THERM is called again for each temperature up to the reservoir temperature, CTAP. With this negative value of CMAXX, all of the temperatures are more than 500°K above CMAXX·CTAP, so that the calculations in THERM are based solely on the thermo fit. The results are stored in TPR(P(I,K,M) for M = 2.

Finally, all of the results for the species are printed out. The properties computed and printed are
Chemical potential in excess of the formation enthalpy, divided by $R_0 T$

$$\text{TPR}_0 (I,1,M) = \frac{H_j^0 - H_{j0}^0}{R_0 T}$$

Molar enthalpy in excess of the formation enthalpy, kcal/mole

$$\text{TPR}_0 (I,2,M) = H_j - H_{j0}$$

Molar heat capacity, cal/mole$^0K$

$$\text{TPR}_0 (I,3,M) = C_{pj}$$

Molar entropy, cal/mole$^0K$

$$\text{TPR}_0 (I,4,M) = S_j^0$$

Examples of the thermal property edit are shown in figures 19 to 46 of Volume II (ref. 2).

### 3.63.2 Transport Property Edit

The transport property edit consists of three parts, the first two of which are always obtained when STUNT2 is called. These parts are (1) an edit of the steps in the transport cross section calculation as specified in the input or in the precoded data; (2) an edit of the "edited" cross section calculations after unneeded steps have been deleted, and needed but unspecified steps have been added on the basis of default options; (3) an edit of the averaged pair cross sections as functions of temperature.

1. **Specified steps in the cross-section calculation.**
   - This first part of the transport property edit is implemented by the D$\phi$ loop ending at statement 150. The index L runs over the steps of the cross section calculation in the order in which they are to be performed. The sequencing array ISEQ(L) selects the steps from the input or precoded data; i.e., the step which is carried out first is the one indexed $I = ISEQ(1)$, etc. NNQI is the number of species pairs to which the cross section calculated in the $L$th step will be applied. The indices of the species belonging to such a species pair are IQJ, JQJ (referred to the
master list of species).* For each step, STUNL2 prints out
L, I, the index KKQ(I) specifying the type of cross section
formula used in the step, the parameter values W(K,I), and
the species names for the first pair of species to which
the results of the step are applied. The names of species
for the remaining species pairs (if any) are then printed
on successive lines.

(2) Steps in the edited cross section calculation.--
The DQ loop ending at statement 190 prints out a description
of the steps in the "edited" cross section calculation which
will actually be used in the transport property calculations
for the current case. This "edited" calculation differs
from the "input" calculation in two major ways: The
steps applying only to species which are absent from the
current gas model have been deleted, and steps which are
required but which were not specified have been provided
on the basis of default options** such as the cross section
mixing rule.

The index L in the DQ loop ending at 190 runs over the
steps in the edited calculation, in the order in which they
are to be performed. KQ(L) is the index of the option
(cross section formula) to be used in the Lth step. NV(KK)
is the number of parameters used by this option. All of
the parameters for all of the steps are stored, without
gaps, in the singly dimensioned array V(I). The parameters
for the Lth step occupy the locations from I = MV1 to MV2
in this array.

The indices of the species belonging to the species
pairs to which the steps of the cross section calculation
apply are stored, without gaps, in two arrays IQ(LQ),
JQ(LQ). These indices IQ and JQ are referenced to the
species list for the current gas model, rather than to the
master list of species. The index LQ for the last species
pair to which the results of the Lth step apply is NQ(L).

*See Section 4.2 of Volume II (ref. 2).

**See Section 4.6 of Volume II (ref. 2).
For each step L, STUNT2 first prints L, the option index KQ(L), and the parameter list V(I) for the option. Then it prints the names, HP(I), HP(J), for the first species pair using a carriage-control character + in column 1 to place these names on the same line. If there are any additional species pairs to which the step is applicable, their names are printed on successive lines.

(3) Averaged transport cross sections. -- If ISW1B is negative, STUNT2 prints a table giving the transport cross sections for the various species pairs as functions of temperature. If ISW1B = -1, STUNT2 also punches these same data on cards for use by other computer programs. Some of the cross sections for charged species depend upon the electron partial pressure as well as the temperature. In STUNT2, the mole fractions SAVEC(I) used by the transport property routines are set to their values in the reservoir (for the current case), and are held fixed as the temperature is varied. The pressure is also held constant at its reservoir value. Thus, all of the cross sections printed are based upon the electron pressure in the reservoir.

The calculations and output are provided by the DO loop ending at statement 230. For each species pair (I,J) with I ≤ J, the temperature is varied from 1000°K to the reservoir temperature CTAP by intervals of 1000°K. At each temperature, subroutine TRANSP is called. With ISW1B < 0, TRANSP does not compute the viscosity, Prandtl number, etc., but does call lower-level transport routines to set up the cross section array Q(L,I,J), where I and J are the species indices and L runs from 1 to 3. These data are printed (and punched for ISW1B = -1), together with the temperature, species indices, species names, and a counter ICARD which serializes the punched cards. Their relation to the cross sections used in transport theory is (Section 3 of Vol. I, ref. 1):

\[
\begin{align*}
Q(1,I,J) &= \overline{\Delta}_{ij}^{(1,1)} \\
Q(2,I,J) &= \overline{\Delta}_{ij}^{(2,2)} \\
Q(3,I,J) &= B_{ij} \overline{\Delta}_{ij}^{(1,1)}
\end{align*}
\]
They are expressed in units of $10^{-16}$ cm$^2$.

### 3.64 Subroutine THERM

THERM is the routine which computes the species thermal properties: $\mu_j^o$ (chemical potential at standard pressure), $H_j$ (molar enthalpy), $S_j^o$ (molar entropy at standard pressure), and $C_{pj}$ (molar heat capacity at constant pressure). In its internal calculations, NATA uses these properties in nondimensional form; the quantities actually computed in THERM are

\[
XMJAT(J) = \frac{\mu_j^o}{R_0 T_0} \tag{167}
\]

\[
SHJ(J) = \frac{H_j}{R_0 T_0} \tag{168}
\]

\[
SENT(J) = \frac{S_j^o}{R_0} \tag{169}
\]

\[
CCPJ(J) = \frac{C_{pj}}{R_0} \tag{170}
\]

where $J = j$ is the species index in the list of species for the current case. In these formulas, $R_0$ denotes the universal gas constant and $T_0$ the reservoir temperature.

When the subroutine is called through the entry THERM, only the heat capacity is computed.

THERM provides two methods for calculating the species thermal properties: the "physical model" and the thermo-fit. These techniques have been explained and discussed in Section 2.2 of Volume I (ref. 1). Data for the physical model are provided for all standard species. In the current version of NATA, thermo-fit data are provided only for the diatomic molecules and molecular ions in air and the planetary atmosphere models. The array IGJ(J) indicates whether thermo-fit data are available for each species. If IGJ(J) = 0, there are no thermo-fit data for the Jth species; in this case the physical model is used.
for the species at all temperatures. If $\text{ETAJ}(J) = 0$, there are no physical-model data, and the thermo fit is used at all temperatures.

In the case of species for which both physical-model and thermo fit data have been provided, the physical model is used from low temperatures up to a nominal switchover temperature, $\text{CTMXXI}$, which is preset to $5000^\circ\text{K}$; and the thermo fit is used at higher temperatures. To prevent possible program failures due to discontinuities in computed species properties resulting from small mismatches between the physical model and thermo fit results, the switchover is spread over a temperature range $1000^\circ\text{K}$ wide, centered at $\text{CTMXXI}$. Equations I(58) are used to mix the results from the two techniques in this temperature range.

The temperature variable used by NATA in its internal computations is $CT$, a nondimensional temperature defined as $T/T_0$. The corresponding nondimensional switchover temperature is $\text{CTMXX} = \text{CTMXXI}/T_0$. For species and temperatures for which the thermo fit is used, the species properties are computed from the formulas I(33), I(34), I(29), and I(31). Prior to the first call to THERM, the thermo fit coefficients $\text{TFA}(J) = a$, etc., have been nondimensionalized in subroutine INIT so as to allow direct use of the nondimensional temperature $CT$ in the thermo fit formulas.

The physical model calculations are done in two steps. The first step computes the translational, rotational, and vibrational contributions to the species properties, and the second step the contributions due to electronic excitation. The first step is coded separately for monatomic and diatomic species, on the one hand, and for linear triatomic species, on the other. In the case of monatomic and diatomic species, the calculations are done using equations I(50), I(51), I(54), I(56), and I(29). For linear triatomic species, equations I(52), I(55), I(57), and I(29) are used.

If the control parameter INEQV is nonzero, the properties of molecular species are calculated on the assumption that the vibrational degrees of freedom are frozen at
the reservoir temperature. In this case, the formulas given in Section 2.4 of Volume I (ref. 1) are used to calculate the species properties.

3.65 Subroutine THRφAT

The function of THRφAT is to adjust the nozzle geometry in the downstream region so as to make the effective area ratio continuous between the upstream solution by the inverse method and the direct downstream solution. It performs this function by multiplying the rescaling factor RSA, which is used in GEφMAR, and which is initially 1.0, by the factor AG/S1. Here S1 is the geometric area ratio based on the specified nozzle geometry before the adjustment, and AG is the geometric area ratio calculated from the effective area ratio AFNX. If the boundary layer is not included in the solution; AG is equal to AFNX. If the boundary layer is included, AG is computed from AFNX by calling subroutine AGSφLN. Since boundary layer displacement effects upon the inviscid flow are neglected during the nonequilibrium solution by the inverse method, the rescaling also provides continuity of the effective area ratio when the coupling of the inviscid flow with the boundary layer is switched on following the call to THRφAT.

In addition to calculating RSA, THRφAT prints an informative message specifying the conditions at the current flow point, and resets the indicator IUPD from 1 to 0 to signify that the switch from the inverse method to the direct integration has occurred.

3.66 Subroutine TRANSP

Transport property calculations are performed in the NATA code under the general supervision of subroutine TRANSP. This subroutine contains two entries: TRANSX which is called to initialize the routines, and TRANSP(TTAB,P) which carries out the actual transport property calculations for any specified gas conditions.

The entry TRANSX is called prior to the start of actual transport property calculations, in order to inform the
routines of the species to be used in the calculations and to perform certain required initializations. The entry first sets the total number of species \( N = I S S \) to be used in the transport calculations and then calls subroutine XSECT which selects cross section data for the species to be used in the calculations from the data available in the code and arranges them in a form convenient for the subsequent computations. Finally, TRANSX calculates a number of quantities depending on the species molecular weights which will be required in the subsequent transport computations, before returning control to the calling program. The quantities calculated here are

\[
ZM1(I) = \frac{128 \times 10^{-24}}{9 \pi N_0 k^3} \quad W_i = 2.8567 W_i \frac{cm^2(\text{OK})^3}{\text{milliwatts}^2(\text{K})^4} \quad (171a)
\]

\[
ZM2(I) = \frac{4}{15 N_0 k} \times 10^7 \quad W_i = 0.032064 W_i \quad \text{g \text{OK}/joule} \quad (171b)
\]

\[
C(I, J) = \frac{8 \times 10^{-12}}{3k} \sqrt{\frac{2 W_i W_j}{\pi N_0 k (W_i + W_j)}} \quad \text{for } I \leq J \quad (171c)
\]

\[
C(I, J) = \frac{W_i - W_j}{W_i + W_j} \quad \text{for } I > J \quad (171d)
\]

Once the gas species have been defined by calling TRANSX, the transport properties for any given gas condition are computed by calling TRANSP(TTAB,P), where TTAB is the gas temperature and P is the gas pressure in atmospheres. The species mole fractions are obtained from the array SAVEC(J) in common block /TEMPRY/. The first step in the computation is the calculation of the molar specific heat for each species in the mixture. This calculation is carried out by calling the entry THERM1 of subroutine THERM. Input to the calculation is by means of the common variable CT which specifies the ratio between the temperature TTAB for which the specific heats are to be calculated and the reservoir temperature CTAP. The nondimensional molar heat
capacities at constant pressure $W_j c_{pj} / N_0 k$ for each species in the mixture are then returned to subroutine TRANSP in the common variable CCPJ(J). These values are saved by the routine in the array DH for later use in computing the internal thermal conductivity from equation I(95), and are also used to compute the nondimensional frozen specific heat\[ CPT = W c_{pf} / N_0 k \]for the mixture, which is required in the calculations of the Prandtl and Lewis numbers, equations I(83) and I(84). In calculating the specific heats, it is necessary to save the previous values of CT and CCPJ in a temporary location and then restore them for later use in other portions of the flow field calculations.

After completing the specific heat calculations, TRANSP next calls subroutine PUTQIN which computes the collision cross sections $\Omega_{ij}^{(1)}$, $\Omega_{ij}^{(2,2)}$, and $B_{ij}^{+} \Omega_{ij}^{(1,1)}$ in units of $10^{-16}$ cm$^2$ for all pairs of species in the mixture at the specified temperature and pressure. These cross sections are returned to subroutine TRANSP in the upper half ($I \leq J$) of the common arrays $Q(1, I, J)$, $Q(2, I, J)$, and $Q(3, I, J)$ respectively. If an edit of pair cross sections has been requested (ISWIB < 0), a RETURN is now executed, and subroutine STUNTS then prints the cross sections contained in the $Q$ array. However, in a normal NATA run (ISWIB = 0), subroutine TRANSP uses these values of the $\Omega_{ij}^{(1,1)}$ to compute the matrix elements $A_{ij}^{(\alpha)}$, $a_{ij}^{(\alpha)}$ and $\Delta_{ij}^{(1)}$, equations I(88) to I(90), which are required in the transport property calculations, and stores them in the $Q$ array, replacing the $\Omega_{ij}^{(1,1)}$ values which were previously stored there. The statements down to 80 compute and store these quantities as follows:

\[
Q(1, I, J) = \frac{1}{k} \sqrt{T} \Delta_{ij}^{(1)} \left( \frac{cm \, (^0K)^{3/2}}{milliwatt} \right) \text{ for } I \leq J \quad (172a)
\]

\[
Q(1, I, J) = \frac{25}{4} \sqrt{T} a_{ij}^{(K)} \left( \frac{cm \, (^0K)^{3/2}}{milliwatt} \right) \text{ for } I > J \quad (172b)
\]

\[
Q(2, I, J) = \frac{25}{8} \sqrt{T} ZM2(I) A_{ij}^{(\mu)} \left( \frac{cm \, (^0K)^{3/2}}{milliwatt} \right)
= \frac{5}{6} \sqrt{T} k \Delta_{ij}^{(2)} \text{ for } I \leq J \quad (172c)
\]
\[ Q(2, I, J) = \frac{25}{8} \sqrt{T} a_{ij}(\mu) \left( \frac{(\text{OK})^{1/2}}{\text{millipoise}} \right) \text{ for } I > J \quad (172d) \]

\[ Q(3, I, J) = \frac{25}{8} \sqrt{T} A_{ij}(K) \left( \frac{\text{cm} (\text{OK})^{3/2}}{\text{milliwatt}} \right) \quad (172e) \]

In connection with these computations, it may be noted that the diagonal matrix elements \(a_{ii}(\alpha)\) are not required in the transport calculations, as shown by equation I(86), so that it is unnecessary to compute them here. Since the quantity \((172e)\) is not symmetric in \(i\) and \(j\), it requires the whole \(Q(3,I,J)\) matrix for storage, whereas the other matrix elements are all symmetric, and can be stored in half the array. The quantity \(\rho c_p D_{ij}/T\) in milliwatts/cm \((\text{OK})^{3/2}\) is also computed and stored by the code as \(\text{FLWIS}\), where \(D_{ij}\) represents the binary coefficient \(I(99)\) between the two species \(ID1\) and \(ID2\) which have been specified for the calculation of the Lewis number in the code input.

The calculation of the transport properties from the matrix elements (172) is begun in the \(D\phi\) loop ending at statement 90, and is then carried forward in a series of three subroutines \(\text{ELCND}, \text{KINT}, \text{and KANDMU}\). The quantities returned to \(\text{TRANS}\) by these subroutines are

\[
\begin{align*}
\text{SIGMA} &= \sigma \text{ in mhos/cm} \\
\text{ZKINT} &= K_{\text{int}}/\sqrt{T} \quad \text{(milliwatts/cm\(-(\text{OK})^{3/2}\))} \\
\text{ZK}(1) &= K_{\text{tr}}/\sqrt{T} \quad \text{(milliwatts/cm\(-(\text{OK})^{3/2}\))} \\
\text{ZK}(2) &= \mu/\sqrt{T} \quad \text{(millipoise/\((\text{OK})^{1/2}\))} \\
\end{align*}
\]

where \(\sigma\), \(K_{\text{int}}\), \(K_{\text{tr}}\), and \(\mu\) are respectively the electrical conductivity, the internal thermal conductivity, the translational thermal conductivity, and the viscosity of the mixture, as given by equations I(85) to I(87), I(95) and I(98). These quantities are then combined further by subroutine \(\text{TRANS}\) according to equations I(83), I(84) and I(94) to obtain the viscosity \(\text{VISC}\) in poise, and the frozen
Prandtl number PRF and the atom-molecule Lewis number* FLEWIS. These quantities, along with the electrical conductivity SIGMA, are then returned to the calling program in the common block TRPROP. For ISWBB ≠ 0, these data, as well as certain other quantities of interest, are also printed out by the final section of subroutine TRANSP before returning control to the calling program.

3.67 Subroutine WEDGE

Subroutine WEDGE calculates the pressure, heat flux, and boundary layer displacement thickness on wedge models. The calculations are done using the modifications I(501), I(503), I(505) of the results of the Cheng-Kemp theory. Optionally, the calculations are also done using the unmodified Cheng-Kemp theory if the input control variable ISW9B is set to a value in the range from -1 to -3.

WEDGE obtains the free-stream flow properties required in the calculations from the array FVOUT in common block /INPUT/. The data in this array refer to the current flow point, because the free stream conditions are always printed out immediately prior to a call to subroutine MODEL which, in turn, calls WEDGE. These data are accessed by equiva-lencing the variable names used in WEDGE to the correspond-ing FVOUT array elements.

The subroutine logic is straightforward. All quantities which are independent of the angle of attack and the wedge leading-edge radius are computed first. The remaining calculations are performed and the results are printed out in a pair of nested DO loops over the index IR for leading edge radii and the index IA for angles of attack.

*The Lewis number as given by I(84) is multiplied by an integer variable IAMBIP, which is equal to 1 if the two species for which the Lewis number is calculated are both neutral, and equal to 2 if one of these species is an ion. The factor of 2 takes account of the ambipolar nature of the diffusion of ions in a neutral plasma.

-192-
For each angle-radius combination, the subroutine performs calculations for all of the specified distances from the leading edge. These distances are specified in the input in two independent ways:

1. A uniform sequence of NWX points, with an initial value of WX1 and an increment DWX;

2. A list of arbitrary values in ascending order.

Subroutine WEDGE folds these two sequences together into a single sequence arranged in ascending order, using logic similar to that employed in subroutine NEXTMP.

WEDGE calls subroutine SHOCK to compute the static pressure ratio across the oblique shock, denoted by APR(IA). This quantity is used in calculating the angle of attack parameter \( \Gamma \) (denoted by CAPGAM) from equation I(501) in the modified Cheng-Kemp theory. After \( \Gamma \) and \( \zeta \) (ZETA) have been computed, WEDGE calls subroutine WESOLN to obtain the approximate solution of the Cheng equation. The arguments of WESOLN are double precision.

The calculations based on the modified Cheng-Kemp theory are performed for ITH = 1. When these calculations have been completed at a given model point, if ISW9B is negative, ITH is set equal to 2 and all of the calculations are done over using the unmodified theory.

The input variable ISW9B is also used to control the output of some results of secondary interest. If IS9=|ISW9B| is equal to 1, the shock ordinate YS is printed; if IS9 is equal to 2, the nondimensional coordinate ZETA is printed; and if IS9 is equal to 3, both YS and ZETA are printed.

3.68 Subroutine WESOLN

The function of subroutine WESOLN is to solve the Cheng equation I(480) for \( z, zz', \) and \( (zz')' \) when \( \zeta' \) and \( \Gamma \) are given. The solution is based on the approximation I(484) which relates the solution for general \( \Gamma \) to that for \( \Gamma = 0. \)
The latter is obtained from Cheng's analytical solution \( I(482), I(483) \) for \( \zeta > 10^{-8} \). For small \( \zeta \), there is a great deal of cancellation of terms in \( I(482) \), so that many significant figures of accuracy are lost. This problem is kept under control by performing all of the arithmetic in WESØIN in double precision, and using an analytical series solution for \( \zeta \leq 10^{-8} \):

\[
z = 1.65096 \, \zeta^{2/3} + 0.50869 \, \zeta^{5/6} - 0.0249 \zeta \quad (174a)
\]

\[
zz' = 1.81711 \, \zeta^{1/3} + 1.25975 \, \zeta^{1/2} + 0.14712 \, \zeta^{2/3} \quad (174b)
\]

\[
(zz')' = 0.6057 \, \zeta^{-2/3} + 0.62988 \, \zeta^{-1/2} + 0.09808 \, \zeta^{-1/3} \quad (174c)
\]

Equation (174a) was derived by Bover and Aiello (ref. 16). Equations (174b) and (174c) can be obtained from (174a) by differentiation.

For \( \zeta > 10^{-8} \), equation \( I(482b) \) is solved for \( \lambda \) using Newton's method. The parameter \( \lambda \) is denoted by XL. An initial trial value of \( \lambda \) is obtained from the rough analytical approximation

\[
\lambda = zz' = (6 \, \zeta^{1/3} + (3 \, \zeta^{1/3}) \zeta^{1/2} \quad (175)
\]

which has a maximum error of about 12 percent. At each step in the Newton iteration, \( \zeta \) is calculated from the current trial value of \( \lambda \) using equation \( I(482b) \), and an improved estimate \( \lambda' \) of \( \lambda \) is obtained from

\[
\lambda' = \lambda - (\zeta - \zeta_i) \frac{d \lambda}{d \zeta} \quad (176)
\]

in which \( \zeta_i \) is the input value ZETA and, from \( I(480) \) with \( I = 0 \) and \( I(483) \),
\[
\frac{d \lambda}{d \zeta} = \frac{d}{d \zeta} \left( z \frac{dz}{d \zeta} \right) = \frac{1 + \sqrt{zz'}}{z} = \frac{1 + \sqrt{\lambda}}{z} \quad (177)
\]

where \( z \) is computed from I(482a). The convergence criterion of the iteration is that the \( \zeta \) from I(482b) equal the input value to within 0.01 percent.

Once the correct value of \( \lambda \) for the solution with \( \Gamma = 0 \) has thus been found, the three outputs of the routine are computed as follows: \( \tau z' \) (denoted by ZZP) is obtained from equation I(491), \( (\tau z')' \) (denoted by ZZPP) from I(434) and I(485), and \( \tau \) (denoted by 2) from I(492).

### 3.6& Subroutine XSECT

Subroutine XSECT provides NATA with the capability to perform transport property calculations for gas models which do not contain all of the species listed in the pre-coded cross section data, and in which the species may be listed in a different order. For this purpose the subroutine sets up a new set of data for the cross section computations in common block /TRANS7/ which gives the same values for the cross sections as the original input data in common block /TRANS2/, but which is in a more condensed format, and is referenced to the set of species included in the current gas model rather than to the larger set of species for which cross sections are defined in the pre-coded and input data. The edited data in common block /TRANS7/ are then used by the code in computing collision cross sections for the gas in the subsequent transport property calculations. The original cross section data are also retained unchanged in common block /TRANS2/, in order to allow the running of stacked cases with different sets of species.

Since several of the options which are used for computing cross sections in NATA (see Section 4.6 of Volume II, ref. 2) give values for the cross sections which depend on the values computed in previous steps of the computation, the edited cross section data in common block /TRANS7/ are set up to carry out the steps of the cross section computations in the same order in which they were specified in the
original data; however, the species references in the data are changed so that the computed cross sections will be indexed in the same way as the species used in the current gas model, rather than being indexed to the original precoded or input data. Further, cross sections for pairs of species not required in the transport calculations are omitted from the edited data, and any missing cross sections are supplied according to the default options described in Section 4.6 of Volume II. The procedures used by subroutine XSECT in preparing these edited data are described in the following paragraphs. It may be noted that several of the cross section options available in the code contain features which require special treatment in this process, so that the coding of the subroutine becomes rather lengthy, although the basic concept is simple. A schematic flowchart of the subroutine showing the major control sections is provided in figure 27.

3.69.1 Initializations

The first executable statement in subroutine XSECT sets the number of species N for which cross sections are to be computed equal to the number of species ISS in the current gas model. The next two statements set IELEC = 1 if electrons are to be considered in the transport calculations and IELEC = 0 otherwise. If electrons are included in the calculations, they must always be the first species.

The statements from "JJ = 50" down through statement 30 establish the correspondence between the species in the current gas model and the master list of species,* which is used in indexing the cross section data supplied in common block /TRANS2/. These statements use the common variable IS(L), which gives the index IS in the master species list of the Lth species in the current gas model (for 1 ≤ L ≤ ISS), to prepare a new array I(J) which gives the position I in the current model of the Jth species in the master species list, for each species included in the current gas model. Species in the master species list which are not included in the current model are filled into the I(J) array in order following the species included in the current model, so that

*See Section 4.2 of Volume II (ref. 2).
Entry

Set N to number of species in gas model

Set I(J) to give correspondence between master species list and species in current gas model

(40) Set up edited cross section data. Revise species references to conform to I(J)

(110) Is a cross section for a species > N required in computations?

no

(300) Delete species > N from edited cross section data

yes

(140) Can this cross section be treated as ≤ N?

no

(190) Revise data to make the cross section ≤ N

yes

(2nd below 150) Increase N by 1; Revise I(J) to include required species in cross section computations

(540) Write diagnostic; terminate case.

no

yes

Is N < 20?

(380) Add Default data for missing cross sections. Can all required data be supplied?

yes

Return

Figure 27. Schematic Flowchart of Subroutine XSECT
I(J) defines a rearrangement of the master species list such that all species J in the original master species list occur at some point I(J) in the rearranged list,* and the first ISS species in the rearranged list are just the species in the current model, in the correct order.

The three statements following statement 30 determine the indices ID1 and ID2 in the rearranged species list of the species to be used in the Lewis number calculations. They are set up so that ID1 < ID2 in all cases.

The common variable Q(1,J,L) is used as a flag in subroutine XSECT to indicate those pairs of species for which cross section data have been supplied. Statements 40 through 50 initialize this variable to 0 for all pairs of species included in the cross section calculations, and the value is then later set to 1 for each species pair for which data are provided. The value Q(1,2,1) = 1. is set initially to indicate that the pair (2,1) is used in the intermediate calculation of the cross sections in subroutine PUTQIN, and is thus not available for use in input. (Note that we are concerned here with ordered pairs (J,L). Only the pairs with J < L are used in the actual transport property calculations,** while those with J > L are available for intermediate calculations if desired.)

3.69.2 Resequencing of Steps and Revision of Species References

The coding following statement 50, down to statement 110, revises the species references in the input cross section data to conform to the rearranged species list I(J) described above and sets up a preliminary version of the edited cross section data in common block /TRANS7/ using these revised data. At this stage of the computation, the

*It is assumed that the master species list contains no more than 50 species; in the current version of NATA, array dimensions limit the number of species to 30.

**See the discussion of subroutine TRANSP in Section 3.66.
edited data contain the same information as the original cross section data in common block /TRANS2/, but the order of steps in the edited data is rearranged to conform to that specified by the sequencing array ISEQ, and the data on species pairs and cross section parameters for each step are close packed in the IQ, JQ, and V arrays. Further, NQ(L) in the edited data is taken to be the sum from 1 to L of the original NQnQ(L) values, and the indices of all species pairs in the edited data are changed to refer to the positions of the species in the rearranged species list, I(J). Since only cross sections Q(J,K) with J ≤ K are used in the transport property calculations, the correspondence between species pairs used in setting up the edited cross section data must preserve this property. Accordingly, the pair (J,K) in the original data is replaced by (I(J), I(K)) if J-K and I(J)-I(K) have the same sign and by the pair (I(K), I(J)) if they have different signs. Since this correspondence is 1-1 between ordered pairs of species, it is clear that when every species pair in the original cross section data is replaced by its corresponding pair in the edited data, the calculated cross section values will all be the same as before, but will be rearranged in the order required for the transport property calculations.

The above correspondence between species pairs is carried out in the NATA code by subroutine CXSECT, which returns the revised indices of a species pair in its argument list when it is called with the original indices. In the coding down through statement 80, this correspondence is used to set up the IQ, JQ list of species pairs for the revised cross section computations. This section of coding also sets Q(1,L,J) = 1 for those species pairs which are included in the IQ, JQ list.

For the cross section options KQ = 9 and 12, there are also references to species pairs in the V parameters of the option which must be revised. This revision is carried out by the statements following statement 80 down through the statement before 110. Subroutine BXSECT searches through the steps L of the cross section computation in common block /TRANS7/, starting with the first step after the value of L specified in its argument, and looking for
a step with $KQ(L) = 9$ or 12. When such a step is found, the subroutine returns the index of the step L, the index $MMV + 1$ of the first V parameter for the step, and the indices LL and J of the species pair which is referenced in the V parameters of the step. Subroutine XSECT then calls CXSECT to revise the indices for the pair and goes on with the search by calling BXSECT again. This process is continued until the final step $L = NKQ$ of the computations is reached; at this point BXSECT sets L to 0 as a signal that the search has been completed and control passes to the next section of code (statement 110). In addition to revising the indexing of the species pairs, this section of code also sets $MMV$, for later use, equal to the total number of V parameters used in the cross section computations.

3.69.3 Deletion of Unused Steps

When control reaches statement 110 of subroutine XSECT, the rearrangement of the species in the cross section data to conform to the order in the current gas model has been completed. The following section of coding, from statement 110 through 380, is now concerned with increasing the efficiency of the cross section computations by omitting unnecessary steps in the computation. In this procedure, the statements from 110 to 300 determine which cross sections are to be omitted from the computations, while those from 300 through 380 carry out the actual editing of the cross section data to delete these values.

As noted previously, the number of species $N$ to be included in the cross section computations is set equal to the number of species in the current gas model at the beginning of subroutine XSECT. Thus only cross sections between pairs of species with indices $I \leq N$ in the revised cross section data are required in the transport property calculations. Ordinarily cross sections involving a species with index $> N$ can be omitted from the cross section computations without changing the values of the computed cross sections for the remaining species with indices $\leq N$; however, for the options $KQ = 9$ and 12 used in the cross section computations, it may happen that the cross section values
for some pair of species with indices \( \leq N \) are to be computed from values obtained previously for some other species pair involving a species with index \( > N \). Statements 110 to 140 check for this possibility, again using subroutine BXSECT to search for those steps in the computation with \( KQ = 9 \) or 12. (Note that the coding here assumes that \( L = 0 \) when statement 110 is reached. This is always true, since statement 110 can only be reached from the IF statements following statements 100 and 290.) If this search does not reveal any pair of species with an index \( > N \) which affects the cross section computations for species with indices \( \leq N \), then it is possible to omit all species with indices \( > N \) from the cross section computations without affecting the values of the calculated transport properties, and control passes at once to statement 300. If, on the other hand, some species pair \((LL,J)\) with an index greater than \( N \) is found which affects the computations for indices \( \leq N \), then the search for further such pairs is temporarily discontinued and the code immediately begins looking for a way to include the pair \((LL,J)\) in the computations. The first approach considered is to treat the cross sections \((LL,J)\) as an intermediate step in the computation of the cross sections for the species with indices \( \leq N \), and to fill the computed cross sections for this pair into some unused location \((II,JJ)\) of the cross section array for species \( \leq N \), where \( JJ < II \leq N \).* Statements 140 to 150 look for such a location, using the previously set array \( Q(I,II,JJ) \) as a flag to indicate which locations are available.

If a suitable location for intermediate storage of the pair \((LL,J)\) is not found in statements 140 through 150, then the following section of code down to statement 190 increases the number \( N \) of species to be included in the cross section computations by one, rearranges the list of species \( I(J) \) for the edited cross section data so that the species corresponding to the maximum index of the pair \((LL,J)\) will be the new species included in the cross section computations (note that by definition this species was not included

*Remember that only species with \( II \leq JJ \) are used in the transport calculations.
in the computations previously), and then returns control to statement 40 to begin the preparation of the edited cross section data again from the beginning, using the new arrangement of species I(J). If this procedure would lead to a value of $N > 20$, however, so that the dimensions allowed for the cross section computations in the code would be exceeded, it is not carried out and control passes instead to statement 540 where a diagnostic message is printed and the error indicator ERR is set to terminate the case. If more than 50 species were specified in the master species list, the case may also be terminated by reaching the GØ TØ 540 statement after statement 180.

When the statements from 140 through 150 are successful in finding an unused location (II,JJ) which can be used for storing the cross sections of the species pair (LL,J), control passes to statement 190, where the value of $Q(1,II,JJ)$ is immediately set to 1.0 to indicate that the pair (II,JJ) is now being used in the cross section computations. The following statements down to statement 500 then revise the edited cross section data in common block /TRANs7/ to change all references to the pair (LL,J) to (II,JJ). To carry out this revision without changing the values of the computed cross sections, any steps in the computation which use the pair (LL,J) in the empirical mixing rule $KQ = 10$ must also be converted in the revision process to use the generalized mixing rule $KQ = 12$. This conversion is carried out by the coding down through the statement NKQ=NSV after statement 270. The code first searches through all steps L of the cross section computations to find any steps using the option $KQ = 10$ which refer to the pair (LL,J). This may occur in two ways: either the species pair (LL,J) may be a non-diagonal pair (i.e., $LL \neq J$) which is computed with the option $KQ = 10$ at some step of the computations, or it may be a diagonal pair (i.e., $LL = J$) which is used in the option $KQ = 10$ to compute the cross sections for some non-diagonal pair involving the species LL. Both of these possibilities are checked for by the IF statement below statement 200, and if either is found the following statements then revise the edited cross section data in common block /TRANs7/ to convert the computation for the given species pair to the option $KQ = 12$. This is accomplished
by inserting the required data for the $KQ = 12$ option into the $KQ$, $NQ$, $IQ$, $JQ$ and $V$ arrays just ahead of the data for the step $L$ which originally computed the cross section, and moving the remaining data in these arrays down to make room for the new data. The original option $KQ = 10$ is retained in the data immediately following the new $KQ = 12$ option; but is no longer applied to that species pair for which the $Q=12$ option is being used. The values of $MMV$, $NKQ$ and $L$ are also adjusted to agree with the revised data. In these computations, subroutine BXSECT is used to locate the position $MM$ in the $V$ array at which the new data are to be inserted. This is done by replacing $NKQ$ by $L$ in common block /TRANS7/, so as to cause BXSECT to end its search of the $KQ$ array, when step $L$ is reached. The true value of $NKQ$ is saved in the variable $NSV$ throughout this section of the code and is restored to its proper location at the end of the section.

After the required conversions from $KQ = 10$ to $KQ = 12$ have been made, control passes to the section of code between card XSE 213 and statement 300, which changes all references to pair $(LL,J)$ in the $IQ$, $JQ$, and $V$ arrays to refer to the pair $(II,JJ)$ instead, using the same algorithms as were used previously in rearranging the species in statements 40 through 110. When these changes have been completed, control returns again to statement 110 and the code again searches the edited cross section data for a pair of species with an index $> N$ which affects the cross section computations for species with indices $\leq N$. If such a pair is found, it is eliminated from the edited cross section data as described above, and the whole process is repeated until either all species pairs with indices $> N$ which affect the cross sections for species $\leq N$ have been eliminated from the data and control passes to statement 300, or else the procedures used in the code are unable to eliminate a species pair and cause the case to be terminated as described above.

By the time control reaches statement 300 in subroutine XSECT, the edited cross section data have been put into such a form that all cross section computations involving species with indices greater than $N$ can be omitted.
without changing the values of the computed cross sections for the species with indices \( \leq N \). The revision of the data to omit these computations is carried out by statements 300 to 380. The pairs of species for which cross sections are computed at each step of the computations are given in the IQ and JQ arrays, so that the steps to be deleted from the computations can be determined simply by searching the IQ, JQ list for pairs in which at least one of the indices IQ or JQ is \( > N \). This search is performed by the preliminary loop ending at statement 310, and the value of IQ for each pair which is to be deleted is set to 0 as a signal for the subsequent calculations.

To determine the required revision of the KQ, NQ, and V arrays in the edited data, it is necessary to establish the correspondence between these arrays and the IQ, JQ arrays. This is done and the arrays are revised in the loop over the steps \( L \) of the cross section computation which ends at statement 370. For each step \( L \) of the computations, the small loop ending at statement 320 determines the correspondence to the IQ, JQ list and calculates the total number of species pairs \( \text{NEWLQ} \) which are not deleted up through step \( L \). This number is the new value of NQ for the step.

The second statement below 330 is the test to determine whether the current step \( L \) of the computations will be omitted from the revised data. The test compares the current value of \( \text{NEWLQ} \) with the value for the previous step; if these two values are equal, it indicates that the current step of the computations will not apply to any species pairs in the IQ, JQ list after the deletions are made, so that the step can ordinarily be omitted from the revised data without affecting the computed cross sections. However, for the option, \( KQ = 10 \) the step is not omitted, even though it does not apply to any species, since it may be required subsequently in supplying the default data for the cross sections, as described below.

The edited cross section data for those steps which are not to be omitted from the computations are placed in their proper locations in the revised KQ, NQ, and V arrays by the coding from the third statement below 330 down to the
statement before 360. Since the revised arrays in this stage of the calculations are always shorter than, or at most equal to, their initial length, this revision can be carried out entirely within the arrays themselves, without disturbing subsequent steps in the revision. The indices L and MV here indicate the positions of the data in the initial arrays, while NEWL and NEWMV are their positions after the desired omissions have been made. It may be noted further in connection with this calculation that no data need be added to the revised V array for KQ = 10, since this option requires no values for V in its operation. Also the number of V parameters NV(K) required by each option is obtained from preset data in the code. For steps which are omitted from the computations, of course, no data need be added to the KQ, NQ, and V arrays; however, the total number of steps NKQ in the computations is reduced by 1 by statement 360.

Once the revision of the KQ, NQ, and V arrays has been completed, the revision of the IQ and JQ arrays is completed in the loop ending at statement 380 by deleting the unwanted species pairs and moving the remaining pairs up to fill in any resulting gaps. This process completes the editing of the cross section data to remove unnecessary steps in the computations which was begun at statement 110 of the subroutine, and provides a set of edited cross section data which will give the same computed cross sections as the original input data for all species included in the current gas model, but which includes only data for the N species for which cross sections are to be calculated, and omits all data for other species which may have been included in the input or the precoded data.

3.69.4 Default Options

The final section of subroutine XSECT, following statement 380, is for the purpose of providing cross section data according to the default options discussed in Section 4.6 of Volume II (ref. 2) for any pairs of species included in the transport calculations for which data were not supplied in the input. These data are added to the edited cross section data in common block /TRANS7/ to provide a complete set of data for the later cross section computations in subroutine PUTQIN.
The first section of coding below statement 380 supplies data for computing the unspecified neutral-neutral cross sections in the gas mixture from the empirical mixing rule, equation I(102). These data are added to the edited cross section data at the last step in which the option $KQ = 10$ is used, or, if this option is not used at all in the data, after the last step of the specified data. This procedure has been adopted in the code to permit data on the resonant contributions to the neutral-neutral cross sections to be specified after the last step in the computations using $KQ = 10$, without affecting the values of the default cross section computed from the mixing rule I(102).

The coding from statement 380 down to statement 400 determines the step $LL$ in the cross section computations at which the unspecified neutral-neutral cross sections are to be added to the data. If no step with $KQ(LL) = 10$ is found in the data, an additional step is added for this purpose at the end, and the value of $NQ(LL)$ for this step is set to $NQ(LL-1)$, since as yet, the step does not apply to any pairs of species in the IQ, JQ list. The statements from 400 through 420 then determine the charge $IZ$, in units of the electronic charge, for each of the species in the transport calculations, using data from the common array $LPIJ$ containing the $\alpha_{ij}$ matrix, and test the value of $IZ$ to determine which are the neutral species. When no electrons are present in the gas, (i.e., $IELEC = 0$), all species are assumed to be neutral and $IZ$ is not set. Since the correspondence $I(J)$ which was previously established between the master species list for the code and the species in the transport property calculations is now no longer needed, the array $I$ is now used to store the indices of the neutral species for later reference. Since no default option has been established in the code for the computation of cross sections for the collisions of a neutral species with itself, the code tests each neutral species to determine whether these data have been specified in the input and if they have not, terminates the case by transferring control to statement 540.
Once the neutral species have been determined, the code calls subroutine AXSECT to add the required data for these species to the edited cross section data in common block /TRANS7/. Subroutine AXSECT then takes the list of neutral species stored in the I array and tests all pairs (J,K) with J < K, using the previously set values of the common variable Q(I,J,K), to determine those pairs for which cross section data have not previously been supplied. All such pairs found in this search are then added to the IQ, JQ list at the LLth step of the cross section computations, the values of the Q(I,J,K) for the added pairs are set to 1.0, and the IQ, JQ, and NQ lists are appropriately revised to take account of these additions. In these calculations subroutine AXSECT assumes that the species to be tested are stored in numerical order in the I array, and that NQ(LL) has been properly set prior to entrance into the subroutine.

After completing the specification of the neutral-neutral cross sections, the code next goes on to add data specifying the unspecified ion-ion cross sections to the edited cross section data. The three statements immediately preceding statement 430 establish a location in the edited data for the insertion of these cross sections. If the last step NKQ in the current version of the data is not presently being used in the cross section computations, it is used for the new data, while if it is being used, an additional step is added to the computations at the end of the current data for the insertion of the new data.

The loop from statements 430 to 480 adds data for computing effective Coulomb cross sections I(100) for the unspecified ion-ion collisions to the edited cross section data. The coding of this loop is set up to cause control to pass through the block of statements from 440 to 460 five times. For the case in which electrons are present (:ELEC = 1), the values of the indices at each passage
through this block are shown in Table III.* For each passage through the block, the loop ending at statement 450 stores some of the species included in the transport property calculations in the array I, the exact species which are chosen depending on the values of the indices for that passage. Thus, referring to Table III, the array I consists of all singly charged ions in the transport calculations for the first passage through the loop, all singly charged ions plus electrons for the second passage, and so forth. The call to XSECT following statement 450 then finds all pairs of the species in the array for which cross sections have not been previously specified, and revises the edited cross section data to include these pairs in the NKQth step of the computations. After returning to subroutine XSECT, the code checks to see whether any unspecified pairs were actually found by subroutine AXSECT in this search, and, if they were, supplies values of the parameters KQ and V for the step which are appropriate for the effective Coulomb potential I(100) from the preset array VCQUL, and then adds another new step at the end of the computations to prepare for storage of the next set of cross section data. The number of entries NEWMV in the V array, which was set originally in the loop ending at statement 340, is also revised to conform to the new data. Thus, the first passage through statements 440 to 460 sets the cross sections for all unspecified pairs of singly-charged ions to the effective Coulomb cross sections given for singly-charged ions in equations I(100a). The

*Note that for IELEC = 0, all species are assumed to be neutral so that all the unspecified cross sections were previously set in treating the neutral-neutral interactions. Thus, no cross sections remain to be set in the present loop and the edited cross section data are accordingly unaffected by the loop, independent of the values of the indices.

-208-
### TABLE III

**INDICES USED FOR CALCULATION OF EFFECTIVE COULOMB CROSS SECTIONS IN SUBROUTINE XSECT**

<table>
<thead>
<tr>
<th>Passage number through</th>
<th>Loop</th>
<th>J</th>
<th>KK</th>
<th>JJ</th>
<th>IE</th>
<th>Species in cross section data are supplied*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(1,1)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>e,1</td>
<td>(e,e) (e,1)</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>(2,2)</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>(1,2)</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>e,1,2</td>
<td>(e,2)</td>
</tr>
</tbody>
</table>

*The notation e indicates electrons, 1 indicates singly charged ions, and 2 indicates doubly charged ions.*
second passage then sets the cross sections for all unspecified pairs involving either singly-charged ions or electrons; but, since the ion-ion cross sections were already set in the previous passage, only electron-electron and electron-ion cross sections are actually set in this passage. From equations I(100a) one sees that the effective Coulomb cross sections for electron-electron and electron-ion collisions are the same (for singly charged ions), so that both types of cross sections can be set in a single step of the computations. Similarly, the last three passages through the loop set the unspecified cross sections involving doubly-charged ions to the values indicated in equations I(100a), as shown in Table III. Thus, when all five passages through the loop have been completed, cross sections will have been specified in the edited cross section data for all pairs of charged species to be included in the transport calculations, up through doubly ionized species. Further, it should be noted that the last step NKQ in the edited data at this stage of the calculations will be an empty position which is not used in specifying the cross sections but is available for the addition of further data.

The preset coding of subroutine XSECT does not provide any default option for specifying the cross sections of ions which are triply charged or higher. Thus the section of code following statement 480 down through the IF statement below statement 490 checks for unspecified ion-ion cross sections for species with higher than double ionization, and, if any are found, terminates the case. Otherwise, the next three statements below the IF statement add all so far unspecified pairs of species in the transport property calculations to the final step NKQ of the cross section computations. Since all neutral-neutral and ion-ion cross sections have previously been specified, the cross sections added at this step are necessarily all neutral-ion cross sections. If any such cross sections are added, parameters are then added to the KQ and V arrays to compute them from an inverse fifth power potential (KQ = 6), using the data on the magnitude of the cross sections stored in the array locations OMEGA1(996), ASTAR (996), and BSTAR(996) in common block/TRANS4/. For the preset data used in the code, this results in the cross sections for the added pairs being set equal to the corresponding N-O⁺ cross
sections. On the other hand, if no unspecified neutral cross sections are found, these data are not added and the value of NKQ is reduced by one to eliminate the unused step at the end of the cross section computations.

The above calculations complete the preparation of the edited cross section data in common block /TRANS7/. The number of steps NKQ in the computations is now checked, and if it does not exceed the maximum allowed by the program dimensions (NKQ = 100), control is returned to subroutine TRANSP. Otherwise the case is terminated by a transfer to statement 540.

In their final version, the edited cross section data prepared by subroutine XSECT specify the procedures to be used in computing all of the cross sections required for the transport property calculations in a given case, and are in a form suitable for later use by subroutine PUTQIN and its associated subroutines in the actual computation of these cross sections. These data are in their final form when control returns from subroutine XSECT, and are not altered by the code in the subsequent transport property calculation; until XSECT is called again to prepare a new set of edited cross section data for another case.
4. GLOSSARY OF FORTRAN SYMBOLS

This section presents a complete glossary of Fortran symbols for the NATA code. Symbols appearing in common are listed first, followed by the remaining symbols in individual routines. The order is as follows: unlabelled common, labelled common in alphanumeric order of the block names, main program, subroutines and functions in alphanumeric order. The variables in each common block are listed in the order in which they appear in the block. The non-common variables are listed in alphanumeric order for each routine. The mathematical notation is as defined in Volume I (ref. 1). The list for unlabelled common includes some variables which are equivalenced to variables in unlabelled common.

4.1 Unlabelled Common

AA(I,J) Matrix of coefficients and constants for a system of simultaneous linear equations

AAA(I,J) Equivalent to AA(I,J)

CDIJ(I,J) Matrix of coefficients $\tilde{\nu}_{i-c,j}$ for expressing the dependent species in terms of the c independent species

CAPX(J) Mole fraction $X_J$ for the Jth species

GJ(J) Species concentrations in moles per gram of mixture; equivalenced to CAPX(J)

A $\alpha$, constant in the density-area relation

AFNTS Effective area ratio calculated for equilibrium flow in the nonequilibrium solution by the perturbation method

AFNX Effective area ratio, $A_e$

AMACH Mach number
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Indicator (initially 0) which is set to 1 when ( dA_n ) becomes positive downstream of the throat in the nonequilibrium solution by the inverse method.</td>
</tr>
<tr>
<td>ARBA</td>
<td>Maximum allowable number of tries at switching from upstream to downstream region in nonequilibrium solution</td>
</tr>
<tr>
<td>ARBB</td>
<td>Counter for number of times upstream-downstream switching point is moved downstream</td>
</tr>
<tr>
<td>BZERG</td>
<td>( b_0/W^0 ), where ( b_0 ) is the covolume of the molecules in the high-density modification of the equation of state and ( W^0 ) is the molecular weight of the undissociated gas.</td>
</tr>
<tr>
<td>C</td>
<td>Constant C in density-area curvefit relation I(383)</td>
</tr>
<tr>
<td>CARB</td>
<td>Effective area ratio at the beginning of the current integration step</td>
</tr>
<tr>
<td>CH</td>
<td>Nondimensional specific enthalpy, ( hW_0/R_0T_0 ), where ( h = ) specific enthalpy, ( W_0 = ) reservoir molecular weight, ( R_0 = ) universal gas constant, ( T_0 = ) reservoir temperature</td>
</tr>
<tr>
<td>CHA</td>
<td>Nondimensional specific enthalpy in the reservoir</td>
</tr>
<tr>
<td>CLNT</td>
<td>( \ln T ), where ( T = ) temperature in °K</td>
</tr>
<tr>
<td>CM</td>
<td>Molecular weight of gas mixture, ( W ) (g/mole)</td>
</tr>
<tr>
<td>CMA</td>
<td>Molecular weight in reservoir, ( W_0 ) (g/mole)</td>
</tr>
<tr>
<td>CRA</td>
<td>( R_0 ), universal gas constant (1.9872 cal/mole °K)</td>
</tr>
<tr>
<td>CRP</td>
<td>( R_0T_0 ), where ( T_0 = ) reservoir temperature (°K)</td>
</tr>
<tr>
<td>CRRB</td>
<td>Intermediate variable in calculation of entropy</td>
</tr>
</tbody>
</table>
CRS  Dummy variable for entropy

CSTA  $0.5 \ln \left( \frac{W_0}{R_0 T_0} \right)$, where $W_0$ is the reservoir molecular weight (g/mole), $R_0$ the universal gas constant, and $T_0$ the reservoir temperature ($^\circ$K)

CT  Ratio of the temperature to the reservoir temperature, $T/T_0$

CTAP  Reservoir temperature, $T_0$ ($^\circ$K)

CTB  Dummy variable for CT

CTC  Dummy variable for CT

CTMAX  Ratio of throat temperature to reservoir temperature

CTMXK  Temperature for switching from thermo-fit to physical model for species properties, divided by reservoir temperature

CTP  Temperature in $^\circ$K

CTPL  $\ln T_0$

CTT  CT at the last printed step of the non-equilibrium solution

CX  Streamwise coordinate $x$(cm), zero at the throat and positive downstream

CXB  CX at the beginning of the current integration step

CXMNAX  Maximum allowable value of $x$(cm)

DATEST  Value of effective area ratio at which nonequilibrium solution is switched from upstream to downstream region

DBTEST  (Not used)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DELT1</td>
<td>Nondimensional temperature decrement used in frozen and equilibrium solutions</td>
</tr>
<tr>
<td>DELT2</td>
<td>(Not used)</td>
</tr>
<tr>
<td>DELTAX</td>
<td>Increment in x (cm)</td>
</tr>
<tr>
<td>DLØGA</td>
<td>$d \ln \Lambda_e / dx$</td>
</tr>
<tr>
<td>DLØGR</td>
<td>$d \ln (\rho / \rho_0) / dx$, logarithmic derivative of nondimensional density</td>
</tr>
<tr>
<td>DT</td>
<td>$d(T_0^0) / dx$, gradient of nondimensional temperature</td>
</tr>
<tr>
<td>ENT</td>
<td>Intermediate variable used in frozen flow solution</td>
</tr>
<tr>
<td>FLUX</td>
<td>Nondimensional mass flux, $(\rho / \rho_0) \cdot u / \sqrt{R_0 T_0 / \bar{W}_0}$</td>
</tr>
<tr>
<td>HDELX</td>
<td>DELTAX/2</td>
</tr>
<tr>
<td>PCT</td>
<td>$\delta (T / T_0)$, perturbation in nondimensional temperature</td>
</tr>
<tr>
<td>PCTEST</td>
<td>Tolerance on $C_\chi$ when testing size of $\delta \chi_i$</td>
</tr>
<tr>
<td>PRES</td>
<td>Nondimensional pressure, $p / p_0$</td>
</tr>
<tr>
<td>PRESA</td>
<td>Reservoir pressure $p_0$ (atm)</td>
</tr>
<tr>
<td>PRESB</td>
<td>Nondimensional pressure at the preceding flow point; used in Mach number calculation</td>
</tr>
<tr>
<td>PRESTH</td>
<td>Nondimensional pressure at throat</td>
</tr>
<tr>
<td>PRHØ</td>
<td>Perturbation in nondimensional density, $\delta \rho / \rho_0$</td>
</tr>
<tr>
<td>RHAP</td>
<td>Density in reservoir, $\rho_0$ ($g/cm^3$)</td>
</tr>
<tr>
<td>Symbol</td>
<td>Definition</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td>RH$\rho$</td>
<td>Nondimensional density, $\rho/\rho_0$</td>
</tr>
<tr>
<td>RH$\rho$B</td>
<td>RH$\rho$ at the preceding flow point; used in Mach number calculation</td>
</tr>
<tr>
<td>RH$\rho$BAR</td>
<td>Effective density in high-density equation of state divided by effective reservoir density</td>
</tr>
<tr>
<td>RH$\rho$C</td>
<td>(Not used)</td>
</tr>
<tr>
<td>RH$\rho$P</td>
<td>Density $\rho$ in gm/cm$^3$</td>
</tr>
<tr>
<td>RHPL</td>
<td>$\ln \rho$</td>
</tr>
<tr>
<td>RHTH</td>
<td>Nondimensional density at throat, $\rho_*/\rho_0$</td>
</tr>
<tr>
<td>R$\rho$BARA</td>
<td>$\tilde{\rho}/\rho_0$</td>
</tr>
<tr>
<td>R$\rho$BARP</td>
<td>Effective density in reservoir, $\tilde{\rho}_0$ (gm/cm$^3$)</td>
</tr>
<tr>
<td>SCPG</td>
<td>$\sum_j \gamma_j C_{pj}/R_0$</td>
</tr>
<tr>
<td>SDT</td>
<td>Change in nondimensional temperature in integration interval</td>
</tr>
<tr>
<td>SEN</td>
<td>Entropy in cal/g oK</td>
</tr>
<tr>
<td>SHPG</td>
<td>$\sum_j \gamma_j H_j/R_0 T_0$</td>
</tr>
<tr>
<td>SC</td>
<td>Factor by which integration interval is increased</td>
</tr>
<tr>
<td>SL</td>
<td>Characteristic length, $l$, now set at 1 cm</td>
</tr>
<tr>
<td>SL64</td>
<td>$1/\sqrt{F}$, where $F$ is the conversion factor from calories to ergs</td>
</tr>
<tr>
<td>SM</td>
<td>Nondimensional mass flux at the throat, RHTH $(SU)_*$</td>
</tr>
<tr>
<td>SU</td>
<td>Nondimensional velocity, $u \sqrt{W_0/R_0 T_0}$</td>
</tr>
</tbody>
</table>
SU2 \quad (SU)^2 \\
SUMG \quad \Sigma \gamma_j \\
TEST \quad \text{Convergence criterion value for Newton-Raphson iterations in calculations of thermochemical equilibrium} \\
TESTB \quad \text{Tolerance allowed in locating temperature at throat for frozen and equilibrium flow} \\
TPRINT \quad \text{Nondimensional temperature interval at which results of nonequilibrium calculation are to be printed} \\
TST\Phi P \quad \text{Minimum value of nondimensional temperature desired in solution} \\
UP \quad \text{(Not used)} \\
ZP \quad \text{Natural log of nondimensional pressure} \\
ZPA \quad \text{Natural log of reservoir pressure in atm} \\
BE(I) \quad \beta_i = \sum_j \beta_{ij} \\
BET(I) \quad -1 + \gamma^{*}_{i-c} \\
BLBK(I) \quad \text{Temporary storage area for boundary layer properties at the switch point from the upstream to downstream solutions} \\
CAI(I) \quad A_i, \text{ constant factor in reaction rate formula} \\
CAPQ(I) \quad \text{Number of gram atoms of the } i^{th} \text{ chemical element per mole of the cold gas; equivalenced to BLBK(I)} \\
CAPXTH(J) \quad X^*_j, \text{ mole fraction of } J^{th} \text{ species at throat} \\
CCI(J) \quad \text{Temporary storage used in calculation of species molecular weights; equivalenced to CAPXTH(J)}
CCP(J)  Molar heat capacity of the $J^{th}$ species divided by $R_0$

CEACT(I)  Activation energy for $i^{th}$ reaction (cal/mole $0^\circ K$)

CGI(I)  Species molecular weights (g/mole)

CGMU(I)  $\ln X_i$, logarithms of mole fractions

CHI(I)  $\chi_i$, variable measuring the departure of the $i^{th}$ reaction from equilibrium, eq. I(289)

CHII(I)  Intermediate variables in computation of equilibrium constant based on mole fractions

CLN1MC(I)  $\ln (1 - \chi_i)$

CLNPI(I)  $\ln P_i$, variables in nonequilibrium solution

CMW(I)  Atomic weights of elements (g/mole)

DGJ(J)  $d \gamma_j/dx$; equivalenced to GJA(J)

ETAI(I)  $\eta_i$, temperature exponent in rate constant for $i^{th}$ reaction

ETAJ(J)  $n_j$, number of atoms in a molecule of the $J^{th}$ species; 0 value indicates no physical model data for thermal properties

GJA(J)  $\gamma_{j0}$, concentration of $j^{th}$ species in reservoir

GJB(J)  GJ(J) at the beginning of the current integration step

PERTGJ(J)  $\delta \gamma_j$, perturbations in species concentrations

PGJ(J)  Intermediate variables in Newton-Raphson calculation of equilibrium composition.
Variables $P_i$ in nonequilibrium solution, eq. I(288)

$P_i \chi_i$

Number of moles $q_j$ of the $j^{th}$ independent species in one mole of mixture

1 if a third-body list is provided for the $I^{th}$ reaction, 0 if not

$\beta_j + (1.5 + n_j) \ln T_0$, constant in physical-model expression for $\mu_j^0$

$\beta_j$, eqs. (51), Vol. I; equivalenced to $SAJ(J)$

$\delta \chi_i$, perturbation in $\chi_i$

Change in $\gamma_j$ over an integration step

Nondimensional species entropy, $S_j^0/R_0$

Nondimensional species enthalpy, $H_j/R_0T_0$

Nondimensional formation enthalpy of $j^{th}$ species, $H_j^0/R_cT_0$

Formation enthalpy of $J^{th}$ species, cal/mole

Intermediate variable in calculation of dependent species concentrations from concentrations of independent species

$\ln \gamma_{j0}$, logarithms of reservoir species concentrations

Temporary storage for quantities necessary to restart a step in the nonequilibrium calculation

$a_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$
TFB(J) $b_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$

TFC(J) $c_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$

TFD(J) $d_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$

TFE(J) $e_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0/T$

TFK(J) $k_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$

THEV(J) Nondimensional characteristic vibrational temperature, $\theta_{vj}/T_0$

THEVP(J) Characteristic vibrational temperature for $J^{th}$ species, $\theta_{vj}$ (°K)

XMJAT(J) $\mu_j^0/R_0T$

XNU1(J) $\nu_i = \sum_j \nu'_{ij}$, sum of stoichiometric coefficients on the reactant side of the $i^{th}$ reaction

BETA(I,J) $\beta_{ij} = \nu'_{ij} - \nu_{ij}$

BTA(I,J) Equivalent to BETA(I,J); not used

ELJ(L,J) Nondimensional energy of $L^{th}$ electronic level of $J^{th}$ species, $E_{lj}/R_0T_0$

GELJ(L,J) Degeneracy of $L^{th}$ electronic level of $J^{th}$ species, $g_{lj}$

XNU1J(I,J) $\nu_{ij}$, stoichiometric coefficient of $J^{th}$ species on reactant side of $i^{th}$ reaction

XNU1JP(I,J) $\nu'_{ij}$, stoichiometric coefficient of $j^{th}$ species on product side of $i^{th}$ reaction

IC Number of ions included in chemical model
IM  Index of the first species in the equilibrium flow calculation, incremented when electrons are dropped from calculation

INEQ  Indicator (initially 0), reset to 1 when numerical integration of nonequilibrium solution is begun

INEQV  Indicator for selecting whether equilibrium (u) or frozen (l) vibrational model is to be used

IP  (Not used)

IRUN  Run number for identification

ISC  Number of elements in mixture (including the electron, if ionized species are included)

ISCPl  ISC+1

ISMC  Number of species minus number of elements (n-c)

ISMCNR  Current value of (n-c) in equilibrium flow calculation (decreased by IC when electrons are dropped from the calculation)

ISR  r, number of reactions

ISS  n, number of species

ISSNR  Current value of n in equilibrium flow calculation (decreased by IC when electrons are dropped from calculation)

ISSP1  n+1

ISSP2  n+2

ISSP3  n+3

ISSP4  n+4
ISW1A  Input control variable; 0 suppresses frozen solution

ISW1B  Input control variable; nonzero value gives edits of transport cross section calculations

ISW2A  Input control variable; 0 suppresses nonequilibrium solution

ISW2B  Input control variable; selects types of input data for reservoir calculation

ISW3A  Input control variable; 0 suppresses equilibrium solution

ISW3B  Input control variable; 0 suppresses boundary layer calculations

ISW4A  Input control variable; must be nonzero if another case follows, 0 for last case

ISW4B  Input control variable; nonzero gives dump in boundary layer routine BLAYER

ISW5A  Input control variable; nonzero gives dump tracing the execution of subroutine RESTMP, which computes the reservoir temperature from mass flow and reservoir pressure or stagnation enthalpy

ISW5B  Input control variable; nonzero gives dump in subroutines EXACT, COMM, NONEQ and PRTA for debugging nonequilibrium calculations

ISW6A  Input control variable; if positive, only the reservoir equilibrium calculation is done; if negative, edit of species thermal properties is produced

ISW6B  Input control variable; 0 suppresses the output of species mole fractions in the free-stream and model-point output
IUPD  Indicator (initially 1), reset to 0 when switch from inverse method to direct integration is made in nonequilibrium solution

IZERØ  0 (set in READ)

JJK  Indicator (initially 0), reset to 1 in NEWRAP when electrons are dropped from the equilibrium calculation

LC  (Not used)

Ml  Index equal to c+l

NFIT  Indicator. 0 if thermo fit data are not used for any species

NIT  Counter for number of iterations in Newton-Raphson procedure

NNN  Integration step counter

NNS  Number of times Δx has been increased by current value of SC

NQS  Number of successful integration steps before Δx is increased

NQT  (Not used)

NTEST  Number of iterations allowed in Newton-Raphson procedure

IGJ(J)  Indicator; 1 if thermo-fit data are used for Jth species, 0 if not

IGM(J)  Number of electronic levels for the Jth species

ITB(I)  Temporary storage for indices necessary to restart step in nonequilibrium calculation
KUR(I,J)  \( U_{ij} \), third body matrix

LPIJ(I,J)  \( \alpha_{ij} \), number of atoms of \( J^{th} \) chemical
          element in a molecule of the \( I^{th} \) species

ACOM(I)  Hollerith description for case

EIMENT(I)  Chemical symbols for elements (Hollerith)

HP(I)  Symbols for species (Hollerith)

4.2 Common /AEGEO/

SQRTA  Square root of the geometric area ratio

S1  Geometric area ratio (reset to the effective
    area ratio in AESOLN)

S2  Derivative of geometric area ratio (reset
    to derivative of the effective area ratio
    in AESOLN)

4.3 Common /AREA/

ATPI(J,I)  Downstream boundary of the \( J^{th} \) section in
          the \( I^{th} \) profile (cm from the throat,
          positive downstream)*

PARAM(L,J,I)  For \( L = 1 \) to 3, the parameter values for the
              \( J^{th} \) section in the \( I^{th} \) profile.  See Sec.
              4.3 of Volume I

RTHCM(I)  Throat radius (cm) for the \( I^{th} \) profile

NSEC(I)  Number of sections in the \( I^{th} \) profile

NSECUT(I)  Number of upstream sections in the \( I^{th} \) profile

ISHAPE(J,I)  Shape index for \( J^{th} \) section in the \( I^{th} \) profile

*"ATPI" is an acronym for "area transfer point input"
NPROFL(I)  Index of \(i\)th profile in the precoded data; NPROFL(1) is equivalent to NOZZLE.

NPRFLS    Number of profiles (1 for a nozzle, 2 for a channel)

NBL       Index (1 or 2) of the profile which diverges from the channel axis least rapidly downstream of the throat

4.4 Common /AVG/

WSAVE     Parameter controlling the averaging distance for the boundary layer correlation parameter, \(n\)

4.5 Common /BL/

DELBL(L)  Displacement thickness of the boundary layer on the \(L\)th profile divided by \(R_0\). For two-dimensional and axisymmetric nozzles, only DELBL(1) is used.

BLINT(L)  Integral \(I\) for the boundary layer on the \(L\)th profile, eq. \(I(172)\)

XZERØ     Position at which boundary layer is assumed to begin (negative value, cm from the throat)

TWALL     Nozzle wall temperature (\(^\circ\)K)

CPWALL    Specific heat of the gas at the wall temperature (cal/g \(^\circ\)K)

VISRØT    Ratio of viscosity in the reservoir to the reservoir temperature (poise/\(^\circ\)K)

DIAM(L)   Throat diameter for the \(L\)th profile (inches)

\(S_W = \left(\frac{h_w}{h_0}\right)-1\), where \(h_w\) is the gas enthalpy at the wall and \(h_0\) the free-stream stagnation enthalpy
RO Characteristic length used in boundary layer calculations (cm); equal to the throat radius for two-dimensional and axisymmetric nozzles; equal to $\sqrt{\frac{A^*}{\pi}}$ for channels, where $A^*$ is the geometric cross sectional area of the channel at the throat.

JDIM Dimensionality index, 0 for two-dimensional nozzles, 1 for axisymmetric nozzles.

IPINT Index of flow solution points at which boundary layer calculations are done.

4.6 Common /BLNE/

XI(I) For $i = 1$ and 2, the streamwise coordinate along the nozzle or channel surface for the $I^{th}$ profile (cm).

ØRDINP(I) The quantity $\phi$ for the boundary layer on the $I^{th}$ profile at the preceding flow point, eq. I(171).

DEBLP(I) $\delta^*/RO$ for the boundary layer on the $I^{th}$ profile at the preceding flow point.

DDBLP(I) $(d \delta^*/dx)/RO$ for the boundary layer on the $I^{th}$ profile at the preceding flow point.

AMPP Mach number at the point preceding the previous flow point.

AMP Mach number at the preceding flow point.

CØUPLD Logical variable, set to .TRUE. when the coupling of the boundary layer to the inviscid flow is switched on.

ISMD The quantity $(30-i)$ in eq. I(418).
### 4.7 Common /BLOUT/

- **REPF**: Reynolds number per foot \( \text{ft}^{-1} \)
- **THETA(L)**: Momentum thickness for the boundary layer on the \( L^\text{th} \) profile, divided by \( R_0 \)
- **SN(L)**: Correlation parameter \( n \), eqs. I(158) and I(174), for the boundary layer on the \( L^\text{th} \) profile
- **XSN(L)**: Averaged correlation parameter \( \bar{n} \), eq. I(217), for the boundary layer on the \( L^\text{th} \) profile
- **PRREF**: Prandtl number at the reference temperature
- **HR**: Recovery enthalpy \( \text{cal/g} \)
- **QWDOT(L)**: Heat flux to the wall surface with the \( L^\text{th} \) profile \( \text{Btu/ft}^2\text{sec} \)
- **TAUWL(L)**: Shear stress on the wall surface with the \( L^\text{th} \) profile \( \text{lb-ft/ft}^2 \)

### 4.8 Common /BLRAD/

- **YOZO**: Product of the two throat radii for a channel \( \text{cm}^2 \)

### 4.9 Common /CERROR/

- **FAILED**: Logical variable, set to .TRUE. when a convergence test on an integration step is failed in subroutine DERIVS or RNKT

### 4.10 Common /CHAN/

- **CPL(I)**: Specifications for standard channel no. 1:
  
  \[ I = 1 \] Index of first channel profile in compiled-in list of profiles

---

-227-
$$I = 2 \quad \text{Index of second channel profile in compiled-in list of profiles}$$

$$I = 3 \quad \text{Hollerith name for channel}$$

$$I = 4 \quad \text{Index (1 or 2) of the profile which diverges from the channel axis least rapidly downstream of the throat}$$

$$I = 5 \quad \text{Hollerith name for facility}$$

- CP2(I) Specifications for standard channel no. 2
- CP3(I) Specifications for standard channel no. 3
- CP4(I) Specifications for standard channel no. 4
- CP5(I) Specifications for standard channel no. 5

4.11 Common /COLDSP/

- CGMW Molecular weight of the cold gas mixture (g/mole)
- QPJ(J) Mole fraction of the Jth species in the cold gas mixture
- IJCS(J) Index of the Jth cold species in the list of species for the current problem
- NCS Number of cold species

4.12 Common /CONVRT/

- CF(1) Conversion factor from nondimensional temperature CT to temperature in °K
- CF(2) Conversion factor from nondimensional pressure PRES to pressure in atm
- CF(3) Conversion factor from nondimensional density RH@ to density in lbm/ft³
CF(4)  Conversion factor from nondimensional velocity SU to velocity in ft/sec

CF(5-7) Conversion factors equal to 1

CF(8)  Conversion factor from viscosity in poise to viscosity in lbm/ft sec

CF(9)  Conversion factor equal to 1

CF(10) Conversion factor from length in cm to length in inches

CF(11) Conversion factor from nondimensional enthalpy CH to enthalpy in Btu/lbm

CF(12) Conversion factor equal to 1

CF(13-16) Conversion factors from nondimensional boundary layer thicknesses DELBL and THETA to thicknesses in inches

CF(17-26) Conversion factors equal to 1

4.13 Common /DRV/

SHJGDGJ $(R_0 T_0)^{-1} \sum H_j \gamma_j / dx$, where $H_j$ is the molar enthalpy of the jth species and $\gamma_j$ is the concentration of the jth species in moles per gram of mixture (set in subroutine EXACT)

4.14 Common /EELEM/

EEPl(J) Data for first input-defined element
   J = 1 Atomic number
   J = 2 Atomic weight (g/mole)

EEPl0(J) Data for tenth input-defined element
   J = 1 Atomic number
   J = 2 Atomic weight (g/mole)
4.15 Common /ELEM/

EP1(I) Specifications for standard chemical element no. 1:

EP1(1) Hollerith name
EP1(2) Atomic weight (g/mole)

EP2(I) Specifications for standard chemical element no. 2

..

EP10(I) Specifications for standard chemical element no. 10

4.16 Common /EQC/

This block is used by subroutine EQCALC to communicate the results of its thermochemical equilibrium calculations to the calling routine.

ZCAP(J) Mole fraction X_j of Jth species

ZSEN Entropy (cal/g °K)

ZCH Nondimensional molar enthalpy, (R_0 T_0)^{-1} \sum X_j H_j

ZCM Mean molecular weight

ZRBP Effective density (\rho_0) in reservoir for imperfect-gas model

ZRH\rho Nondimensional density, \rho/\rho_0

4.17 Common /EQC2/

ZPZ ln (p/p_0), logarithm of the nondimensional pressure

ZGMU(I) ln X_i, logarithm of the mole fraction for the Ith chemical species

-230-
4.18 Common /ERROR/

ERR Logical flag set to "true" in DUMP routine; if "true", control is passed back to Main program, DUMPEX is called to print a dump of common data, and case is terminated

4.19 Common /GLIM2/

NØREAC(I) Logical control variables; when NØREAC(I) is set to .TRUE., the Ith reaction is suppressed

4.20 Common /INGNE/

AIN(K,I) The inverse $A_{ki}$ of the square submatrix $\Xi_{ij}$ of the matrix $\Xi_{ij}$ specifying the number of atoms of the jth element per molecule of the ith species; see eq. 1(3)

4.21 Common /LN/

ISATØM Index of atom used in Lewis number calculation, in master list of species

ISMØL Index of molecule used in Lewis number calculation, in master list of species

JATØM Index of atom used in Lewis number calculation, in list of species for current problem

JMØL Index of molecule used in Lewis number calculation, in list of species for current problem

4.22 Common /MASSFL/

SMASS Sonic mass flux based on total mass flow and geometric throat area (g/cm²sec)

CTMXXI Temperature (°K) above which species thermal properties are computed from the thermo fit for those species for which thermo fits are supplied

-231-
TSTØPI  Free-stream temperature at which the flow solutions will be terminated (°K)

IS(J)  Index of the Jth species in the current problem, in the master list of species

4.23 Common /MIXT/

GP1(I)  Specifications of standard gas mixture no. 1:

I = 1  Hollerith name of mixture

I = 2  Number of chemical elements in mixture, including e⁻ if the gas model contains ion species

I = 3  Number of chemical species in mixture, including e⁻ if model contains ion species

I = 4  Number of reactions included in gas model

I = 5  Number of ions in gas model, excluding e⁻

I = 6-15  Indices of elements present in mixture, in master list of elements; if electrons are present, they are the first element

I = 16-25  Mole fractions (QPJ) of the cold species

I = 26-45  Indices of the species included in the gas model, in the master list of species

I = 46-109  Indices of the reactions included in the model, in the master list of reactions
I = 110-119 Indices of the cold species, in the master list of species

I = 120 Number of cold species

I = 121 Atom index for Lewis number calculations, in the master list of species

I = 122 Molecule index for Lewis number calculations, in the master list of species

I = 123 INT. If INT = 0, electron temperature equals gas temperature. If INT > 0, the model includes electronic nonequilibrium, and INT is the index of the extra reaction properties required in TNEP(I,INT)

I = 124 Indicator for inclusion (1) or exclusion (2) of Fay-Riddell Lewis number factor in the stagnation-point heat flux

GP2(I) Specifications of standard gas mixture no. 2

GP6(I) Specifications of standard gas mixture no. 6

4.24 Common/ MODPAR/

XMP1 Initial distance from throat at which model condition calculations are to be done (cm)

DXMP (Not used)

FSTAG Control variable; 0. value suppresses frozen shock calculations at model points, negative value suppresses equilibrium shock
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CATFAC</td>
<td>Catalytic efficiency parameter $\theta$ for stagnation point heat flux calculations; 0 for noncatalytic wall</td>
</tr>
<tr>
<td>TMODEL</td>
<td>Model wall temperature for stagnation point heat flux calculations (°K)</td>
</tr>
<tr>
<td>XMODEL</td>
<td>Initial distance from the throat at which model condition calculations are to be done (inches)</td>
</tr>
<tr>
<td>DXMODEL</td>
<td>(Not used)</td>
</tr>
<tr>
<td>TPLATE</td>
<td>Wall temperature for calculations of heat flux to a flat plate at zero angle of attack (°K)</td>
</tr>
<tr>
<td>KDIM</td>
<td>Control variable for stagnation point heat flux calculations; 0 for two-dimensional model geometry, 1 for axisymmetric model</td>
</tr>
</tbody>
</table>

4.25 **Common /MODPT/**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSDIAM(I)</td>
<td>Test-section diameters at which model condition calculations are to be done (inches); in channel flow problems, specified channel widths at which flow calculations are to be done</td>
</tr>
<tr>
<td>TSAR(I)</td>
<td>Geometric area ratios at the nozzle stations defined by TSDIAM(I)</td>
</tr>
<tr>
<td>NTS</td>
<td>Number of specified test section diameters</td>
</tr>
<tr>
<td>MBL</td>
<td>In channel flow problems, index of the profile defining the channel width</td>
</tr>
</tbody>
</table>

4.26 **Common /MODP2/**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODLP</td>
<td>Logical indicator, .TRUE. when the current step in the nonequilibrium solution is at a model point</td>
</tr>
</tbody>
</table>
4.27 Common /NEQ/  

ØMDST(L) \[1 - \delta^*/R0\] for the Lth profile at the sonic point  

DDELBL(L) \[(d \delta^*/dx)/R0, \text{derivative of nondimensional displacement thickness for the boundary layer on the Lth profile}\]  

4.28 Common /NEWMP/  

FACMP Factor by which distance from throat is increased in a geometric sequence of model points  

NMØDPT Number of model points in geometric sequence  

4.29 Common /NØZZ/  

ZPl(i) Specifications of standard nozzle profile no. 1:  

I = 1 Throat radius in cm  

I = 2 Distance upstream of the throat of the point at which the boundary layer is assumed to originate (negative value, cm)  

I = 3 Number of profile sections upstream of the throat  

I = 4 Number of profile sections downstream of the throat  

I = 5-16 Shape indices (ISHAPE) of the profile sections (in order from upstream to downstream):  

ISHAPE = 1 Straight section  

ISHAPE = 2 Circular section convex toward axis  

-235-
ISHAPE = 3  Circular section concave toward axis

I = 1.7-27  Downstream boundaries of profile sections (distances from the throat, positive downstream, cm)

I = 28-63  Parameters defining the profile sections (lengths in cm).* There are three parameter values P1, P2, P3 for each profile. For ISHAPE = 1, the equation of a straight profile is

\[ r(x) = P_1 + P_2x \]

For ISHAPE = 2 or 3

\[ P_1 = \text{distance of circle center from axis} \]
\[ P_2 = \text{x-coordinate of circle center} \]
\[ P_3 = \text{circle radius} \]

I = 64  Hollerith facility name

ZP2(I)  Specifications of standard profile no. 2

\[ \cdot \]
\[ \cdot \]

ZP20(I)  Specifications of standard profile no. 20

4.30  Common /QUIMID/

TB7  \[ N_0 \rho_0, \text{ where } N_0 \text{ is Avogadro's number (6.0225 x } 10^{23} \text{ particles per mole) and } \rho_0 \text{ is the density in the reservoir in } g/cm^3. \]

*See Section 4.7 of Volume II
4.31 Common /OUTPUT/

FVOUT(I) Array used for output of flow variables; set in subroutine OUT1 (entry OUT2)

I = 1 Axial coordinate, x (inches)
I = 2 Temperature (°K)
I = 3 Static enthalpy (Btu/lb)
I = 4 Pressure (atm)
I = 5 Density (lb/ft³)
I = 6 Nozzle diameter (inches); geometric area ratio in channel solutions
I = 7 Velocity (ft/sec)
I = 8 Mach number
I = 9 Entropy (Btu/lb°R)
I = 10 Frozen specific heat ratio
I = 11 Effective area ratio
I = 12 Reynolds number (per ft)
I = 13 Molecular weight (g/mole)
I = 14 Viscosity (lb/ft sec)
I = 15 Electrical conductivity (mho/cm)
I = 16 Geometric area ratio; width of face with first profile in channel solutions
I = 17 Displacement thickness (inches)
I = 18 Momentum thickness (inches)
I = 19  Heat flux to wall (Btu/ft²sec)
I = 20  Shear stress on wall (lbf/ft²)
I = 21  Recovery enthalpy (Btu/lb)
I = 22  Prandtl number at the reference temperature
I = 23  Stanton number
I = 24  Reynolds number based on θ
I = 25  Re_θ for boundary layer transition
I = 26  Width of face with second profile (channel solutions only)
I = 27  Displacement thickness on second profile (inches)
I = 28  Momentum thickness on second profile (inches)
I = 29  Heat flux to wall on second profile (Btu/ft²sec)
I = 30  Shear stress on wall on second profile (lbf/ft²)
I = 31  Recovery enthalpy (Btu/lb)
I = 32  Prandtl number at the reference temperature
I = 33  Stanton number on second profile
I = 34  Re_θ on second profile
I = 35  Re_θ for transition

GJMF(J)  (Not used)
4.32 Common /POLYAT/

THEVE(K,J) For K = 1-4, the four characteristic vibrational temperatures of the Jth species in the current gas model, if this species is a linear triatomic molecule. In subroutine READ, the entries K = 2-4 are set to values in °K. In subroutine INIT, these values are nondimensionalized by dividing them by the reservoir temperature, and the value K = 1 is set.

4.33 Common /RDLIST/

IGAS Index of standard gas mixture
IGASØ IGAS value in preceding case (if any)
NOZZØ Nozzle profile index for preceding case
ICHANO Channel index for preceding case

4.34 Common /RDMAIN/

HS Input value of stagnation enthalpy (cal/g)
SUPGØ Logical control for suppressing output of the gas model data when the model is the same as in the preceding case

MFITER Input control for the iteration to take the displacement thickness into account in the reservoir condition calculations; zero value suppresses the iteration

NORMAN Input logical control for suppressing all transport property calculations
4.35 **Common /RDMOD/**

**LEWIS**
Index controlling use of the Fay-Riddell Lewis number factor in stagnation-point heat flux calculations. For LEWIS = 1, the factor is used; for LEWIS = 2, it is omitted.

**IAMBIP**
Index specifying whether the diffusion process upon which the Lewis number is based is ambipolar (IAMBIP = 2) or not (IAMBIP = 1).

4.36 **Common /RDOUT/**

**FLW**
Total mass flow in lbm/sec

**FACNAM**
Hollerith facility name

**CHANAM**
Hollerith channel name

**LIMOUT**
Index, nonzero if the boundary layer parameters N and XSN are to be printed in the output, equal to 0 if not.

4.37 **Common /RDTR/**

**ISWBB**
Index controlling diagnostic dumps in the transport property routines. If ISWBB = 0, these dumps are omitted. If ISWBB > 0, the PUTQIN dump is produced once every ISWBB times the subroutine PUTQIN is called. If ISWBB < 0, the PUTQIN dump is suppressed.

4.38 **Common /RDWEDG/**

**ANGLE(I)**
Angles of attack (degrees)

**RADLE(J)**
Radii of leading edge (inches)

**WX1**
Initial distance from leading edge of wedge (inches)

**DWX**
Increment in distance from leading edge (inches)
WXI(K)  Specified distances from leading edge (inches)
TWEDGE  Surface temperature of wedge model (°K)
WK      Leading-edge drag coefficient, k
NWX     Number of distances from leading edge (at increment of DWX) at which surface conditions on wedge model are to be calculated
NANGLE  Number of angles of attack
NRADLE  Number of leading-edge radii
WEDGEM  Logical control variable; if "false", conditions on wedge are not calculated
AXISYM  Logical control variable; if "false", stagnation point heat fluxes are not calculated
ISW9B   Control variable, normally 0, with the following possible optional values:
+ 1  Print shock ordinate Ys
+ 2  Print nondimensional coordinate \( \xi \)
+ 3  Print both \( Y_s \) and \( \xi \)
If ISW9B is negative, calculations are done using the unmodified Cheng-Kemp theory as well as the modified theory.

4.39 Common/REAC/

RPL(I)  Specifications of standard reaction no. 1:

I = 1    Coefficient A in formula for forward reaction rate constant (cm³/mole sec or cm⁶/mole² sec)
I = 2    Exponent \( \eta \) in formula for rate constant
I = 3 Activation energy $E_a$ in formula for rate constant (cal/mole)

I = 4 Equal to 1.0 if a third-body list is provided for the reaction, equal to 0.0 if not

I = 5 Number of reactant species ($\leq 3$)

I = 6 Number of product species ($\leq 3$)

I = 7-9 Indices of reactant species in master list of species

I = 10-12 Indices of product species in master list of species

I = 13-15 Numbers of molecules of reactants

I = 16-18 Numbers of molecules of products

I = 19 Number of third bodies

I = 20-29 Indices of third-body species in master list of species

RP2(I) Specifications of standard reaction no. 2

RP92(I) Specifications of standard reaction no. 92

4.40 Common /READAT/

PRESAI Reservoir pressure input (atm)

DELT1I Input value of nondimensional temperature decrement used in the frozen and equilibrium flow calculations and in starting the non-equilibrium solution
TPRNTI  The free-stream nonequilibrium solution is printed out at temperature intervals greater than or equal to TPRNTI times the reservoir temperature. For TPRNTI = 0., every step is printed.

DELTXI  Input value of initial step size in x for nonequilibrium integration (cm)

CXMAXI  Distance beyond the throat at which the flow solutions are stopped (inches)

CTAPI  Reservoir temperature input (°K)

XZEROI  Nozzle or channel inlet position at which the boundary layer is assumed to originate (negative value, measured in inches upstream from the throat)

PARAMI(L,M,N) Input array of parameter values for nozzle or channel profiles (see definitions of code inputs, Group 4, in Section 2.3 of Volume I)

BZEROI  Input of constant in imperfect IPS correction

HSTAG  Input stagnation enthalpy (Btu/lb)

READXS  Input control for reading cross section data for transport property calculations

READG  Input control for reading data on elements, species, and/or reactions

AAMS  Input control for automatic air model selection

AXIMOÐ  Input control for suppressing stagnation point model condition calculations if only wedge calculations are desired

ICASE  Case number in the current job

NEELS  Number of elements being defined in the input
ISW7B  Input control for suppressing the output of the boundary layer parameters N and XSN
INT   Indicator for electronic nonequilibrium in a gas model
ICHAN Index of standard channels
NQSI  Number of successful integration steps required before the step size in the nonequilibrium calculation is increased
INEQVI Input control for equilibrium or frozen molecular vibration
NRECO Input of number of records already on data tape at beginning of run
JCS(I) Indices of cold species in master list of species
ISCI  Input of the number of chemical elements in the gas model
ISSI  Input of the number of chemical species in the gas model
ISRI  Input of the number of reactions in the gas model
ICI   Input of the number of ions in the gas model
IE(I) Input indices of elements present in the gas model, in the master list of elements
IR(I) Input indices of reactions in the gas model, in the master list of reactions
NSECTS(I,J) For I = 1, number of upstream sections in the curvefit for the Jth profile; for I = 2, number of downstream sections
IEEP(I) Indices assigned to the input-defined elements in the master list of elements
4.41 Common /RENE/

GAMIN  Concentration (moles/g) below which a species will be frozen if it decreases so rapidly that it controls the integration step size.

HTEST  Maximum relative change in total enthalpy in an integration step.

TETEST  Maximum relative change in electron temperature in an integration step

QTEST  Criterion value for maximum allowable change in energy transfer to electron gas in an integration step

DCHLL  Parameter used in selecting initial integration step size; the initial $\Delta x$ is not allowed to exceed $0.01 |\Delta \chi_i^{'}|_{\text{min}}/\text{DCHLL.}$

DCHRAT  Parameter controlling the artificial increase in reaction rates in the perturbation solution to avoid premature startup cf the numerical integration (Section 7.3.7 of Volume I)

CCHI  Parameter $C_\chi$ in criterion for switch from perturbation solution to numerical integration

TTEST  Maximum relative change in the gas temperature in an integration step

GTEST  Maximum relative change in a species concentration in an integration step

4.42 Common /RESPRP/

VISCR  Viscosity in the reservoir (poise)

PRR  Prandtl number in the reservoir

SIGR  Electrical conductivity in the reservoir (mho/cm)
FLEWR

Lewis number in the reservoir

4.43 Common SPEC

SP1(I)

Specifications for standard species no. 1:

I = 1 Hollerith name for species
I = 2 Number of elements in species (≤ 3)
I = 3-5 Indices of elements in the master list of elements
I = 6-8 Numbers of atoms of elements
I = 9-14 Thermo-fit parameters a, b, c, d, e, k
I = 15 Formation enthalpy (cal/mole)
I = 16 Number of atoms in a molecule of the species
I = 17 Chemical constant, b
I = 18 Characteristic vibrational temperature (°K)
I = 19 Number of electronic levels (≤ 10)
I = 20 1 if thermo-fit is used, 0 if not
I = 21-30 Degeneracies of electronic levels
I = 31-40 Energies of electronic levels (cal/mole above ground state)
I = 41-43 Second, third, and fourth characteristic vibrational temperatures for linear triatomic species (°K)
4.44 Common /SS/

CAS  Square of the sound speed divided by the non-dimen-
      sional temperature $T/T_0$ (cm$^2$/sec$^2$)

US   Standard velocity used for nondimensionalizing
      the flow velocity, $\sqrt{R_0 T_0/W_0}$, where $T_0$ is
      the reservoir temperature, $W_0$ the reservoir
      molecular weight, and $R_0 = 8.31436 \times 10^7$
      erg/mole$^0$K

4.45 Common /STAG/

T2E  Initial estimate of temperature behind
      equilibrium normal shock ($^O$K)

EPLSN Initial estimate of density ratio across
      equilibrium normal shock

T2F  Initial estimate of temperature behind frozen
      normal shock ($^O$K)

EPSF Initial estimate of density ratio across
      frozen normal shock

4.46 Common /STEPS/

NSTEPS Number of successful integration steps since
         the last printout of conditions at a flow
         point

4.47 Common /SWITCH/

XPB  The corrected x-coordinate of the flow point
      preceding the point at which BLAYER is first
      called with x corrected for the displacement
      thickness in the frozen and equilibrium
      solutions (cm)

DXPB The difference in corrected x values between
       the point at XPB and the preceding point
Constant parameter \( w \) determining the characteristic distance over which the boundary layer correlation parameter \( n \) is averaged; see eqs. I(218) and I(217)

4.48 Common /TAP\( \text{OUT} /\)

XXX(I) Block of storage used to set up data which are to be saved on binary tape for subsequent processing by other programs

ITP\( \text{OUT} \) File number of binary output tape

NRC\( \text{OUT} \) Number of records written on binary tape during current case of job

IFL\( \text{OW} \) (Not used)

ITYPER Type of record written on binary tape:

1 "Case" record with overall definition of flow solution being computed

2 "Point" record containing free-stream flow variables

3 "Model" record containing data for stagnation point model

IMP Index of model points written onto binary tape in current solution

DATAPE Logical control variable; if "false", no data are written onto binary tape

4.49 Common /TEMP\( \text{RY} /\)

SAVEC(J) Species mole fractions for which transport property calculations are done

4.50 Common /TH\( \text{RT} /\)

RSA Factor by which cross sectional area of flow has been rescaled in nonequilibrium solution
### 4.51 Common /TNCE/

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMGH</td>
<td>$\sum_j \gamma_j$, sum of the molar concentrations for all species except the electron (mole/g)</td>
</tr>
<tr>
<td>SCPGH</td>
<td>$\sum_{j=1}^{n} C_{pj} \gamma_j / R_0$, sum of the products of the molar heat capacities and molar concentrations for all species except the electron, divided by the universal gas constant</td>
</tr>
<tr>
<td>QDPR</td>
<td>Power radiated from the gas, cal/cm³−sec</td>
</tr>
<tr>
<td>QDPE</td>
<td>Net power transferred to the electron gas (cal/cm³−sec)</td>
</tr>
</tbody>
</table>

### 4.52 Common /TNDC/

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMJATD(J,L)</td>
<td>Chemical potential at standard pressure ($\mu_i^0$; for the $i$th species, evaluated at the $L$th temperature ($L = 1$, heavy particle temperature; $L = 2$, electron temperature), divided by $R_0$ times the $L$th temperature</td>
</tr>
<tr>
<td>CLNTD(L)</td>
<td>Natural logarithm of the $L$th temperature ($L = 1$, heavy particle temperature; $L = 2$, electron temperature)</td>
</tr>
<tr>
<td>CTD(2)</td>
<td>Nondimensional temperatures ($L = 1$, $T/T_0$; $L = 2$, $T_e/T_0$)</td>
</tr>
</tbody>
</table>

### 4.53 Common /TNE/

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNL(I)</td>
<td>Specifications of data for standard electronic nonequilibrium model no. 1 (helium): $I = 1-25$ KTF(I), indicator for type of formula for forward rate constant of the $I$th reaction; see Sec. 4.4 of Vol. II</td>
</tr>
</tbody>
</table>

-249-
I = 26-50  KTR indicators for the reverse reactions in the gas model. 

KTR = 0 if the backward rate constant $k_r$ is zero; KTR = 1 if $k_r = k_r(T)$; and KTR = 2 if $k_r = k_r(T_e)$

I = 51-75  ITR values for the reactions in the gas model. ITR is an indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses. The significance of its values is discussed in Sec. 4.4 of Vol. II

I = 76-100  $\epsilon_0$ parameters for reactions

I = 101-125  Values of the parameter "$a" for reactions with ITR = 1

I = 126-155  Temperature values for table of elastic collision cross section, $Q^{(1,1)}$

I = 156-185  $Q^{(1,1)}$ values for table

I = 186  Parameter b for reactions with KTF = 4

TN2(I)  Specifications of data for electronic non-equilibrium model no. 2 (argon)

4.54 Common /TNEQ/

TLIST(I)  For I = 1-30, temperatures for elastic collision cross section table for current gas model (OK)

POM(I)  For I = 1-30, cross section values for elastic collision cross section table for current gas model

-250-
4.55 Common /TNERK/

SDTE $\Delta T_e / T_0$, change in electron temperature in integration step

CTEB $T_e / T_0$ at the start of the integration step

DCHA $(W_0 / R_0 T_0) (dh_0 / dx)$, rate of change of non-dimensional total enthalpy

CHB $W_0 h_0 / R_0 T_0$, nondimensional total enthalpy at start of integration step

SDCHA $\Delta (W_0 h_0 / R_0 T_0)$, change in nondimensional total enthalpy in integration step

DQMAX $D_{gm}$, criterion value for convergence test on changes in energy transfer to the electron gas

IFAIL Indicator for cause of integration step failure:

<table>
<thead>
<tr>
<th>IFAIL</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Failure in COMM or DERIVS (called from RNKT)</td>
</tr>
<tr>
<td>1</td>
<td>Test on T in NØNEQ</td>
</tr>
<tr>
<td>2</td>
<td>Test on $T_e$ in NØNEQ</td>
</tr>
<tr>
<td>3</td>
<td>Test on $h_0$ in NØNEQ</td>
</tr>
<tr>
<td>4</td>
<td>Test on DL$\omega$GA in NØNEQ</td>
</tr>
<tr>
<td>5</td>
<td>Test on QDPE in NØNEQ</td>
</tr>
<tr>
<td>11-30</td>
<td>Test on GJ(IFAIL-10) from RNKT in NØNEQ</td>
</tr>
<tr>
<td>31-50</td>
<td>Test on GJ(IFAIL-30) from element conservation in NØNEQ</td>
</tr>
<tr>
<td>IFAIL</td>
<td>Meaning</td>
</tr>
<tr>
<td>---------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>60</td>
<td>Failure in DERIVS or COMM (called from NOEQ)</td>
</tr>
<tr>
<td>-1 to -ISS</td>
<td>Test on sign of GJ(-IFAIL)</td>
</tr>
<tr>
<td></td>
<td>in RNKT (GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+1)</td>
<td>Test on sign of T in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+2)</td>
<td>Test on sign of T in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+3)</td>
<td>Test on sign of h0 in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+4)</td>
<td>Test on QDPE in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ2 or GJ3)</td>
</tr>
<tr>
<td>-31 to -30+ISS</td>
<td>Test on sign of GJ(-IFAIL-30)</td>
</tr>
<tr>
<td></td>
<td>in RNKT (GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(31+ISS)</td>
<td>Test on sign of T in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(32+ISS)</td>
<td>Test on sign of T in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(33+ISS)</td>
<td>Test on sign of h0 in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(34+ISS)</td>
<td>Test on QDPE in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-61 to -60+ISS</td>
<td>Test on sign of GJ(-IFAIL-60)</td>
</tr>
<tr>
<td></td>
<td>in RNKT (GJ4, Treanor)</td>
</tr>
<tr>
<td>-(61+ISS)</td>
<td>Test on sign of T in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ4, Treanor)</td>
</tr>
<tr>
<td>-(62+ISS)</td>
<td>Test on sign of T in RNKT</td>
</tr>
<tr>
<td></td>
<td>(GJ4, Treanor)</td>
</tr>
</tbody>
</table>
IFAIL

\(-\left(63+\text{ISS}\right)\) Test on sign of \(h_0\) in RNKT (GJ4, Treanor)

\(-\left(64+\text{ISS}\right)\) Test on QDPE in RNKT (GJ4, Treanor)

4.56 Common /TN\$\$NEQ/

CTE Electron temperature divided by gas temperature in reservoir

DTE Derivative of CTE with respect to \(x\), i.e., \(d(T_e/T_0)/dx\)

BPAR Parameter \(b\) for reactions with \(KTF = 4\)

EPAR(I,JR) \(\text{EPAR}(1,JR) = \text{parameter } \epsilon_0\) for the \(J_{Rth}\) reaction in cal per \(N_0\) reactions; \(\text{EPAR}(2,JR) = \text{parameter } a\) for the \(J_{Rth}\) reaction if \(\text{ITR}(JR) = 1\)

NT Indicator for electronic nonequilibrium in the gas model; for \(NT = 1\), the gas and electron temperatures are assumed to be equal; for \(NT = 2\), they are not

ITR(JR) Indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses in the \(J_{Rth}\) reaction of the current gas model. See discussion in Section 4.4 of Vol. II

KTF(JR) Indicator for temperature dependence of the forward rate constant \(k_f\) for the \(J_{Rth}\) reaction in the current gas model; see Sec. 4.4 of Vol. II
KTR(JR) Indicator for temperature dependence of the backward rate constant \( k_r \) for the JRth reaction in the current gas model:

\[
\begin{align*}
KTR = 0 & \quad k_r = 0 \\
KTR = 1 & \quad k_r = k_r(T) \\
KTR = 2 & \quad k_r = k_r(T_e)
\end{align*}
\]

ICH(J) Indicator for charged species; equal to 1 if the Jth species is neutral, equal to 2 if the species is an ion.

IPA(I) Species index for the product atom in the Ith reaction; if \( KTF(I) \) is not equal to 4, IPA(I) is set to zero.

4.57 Common /TRAN\( I \)/

IIIi(K) For \( K = 1 \) to a maximum of 5, the index of the first species in the Kth pair of species to which the cross section calculated in the \( i \)th step of the transport property calculation is applied. The common block provides storage for \( i = 1 \) to 100.

4.58 Common /TRAN\( J \)/

JJi(K) For \( K = 1 \) to a maximum of 5, the index of the second species in the Kth pair of species to which the cross section calculated in the \( i \)th step of the transport property calculation is applied. The common block provides storage for \( i = 1 \) to 100. Only pairs with \( IIIi(K) \leq JJi(K) \) are used.

4.59 Common /TRAN\( S \)/

T Temperature in \( ^{\circ}\)K for transport calculations.
$Q(K,I,J)$ is used in the cross section computations to store the averaged cross sections for the species pair $(I,J)$ as follows:

\[
Q(1,I,J) = \overline{N}_{ij}^{(1,1)},
\]
\[
Q(2,I,J) = \overline{N}_{ij}^{(2,2)},
\]
\[
Q(3,I,J) = B_{ij}^{*} \overline{N}_{ij}^{(1,1)}.
\]

At other stages in the calculations, $Q(K,I,J)$ is also used in subroutine TRANSP to store the matrix elements for the transport calculations, and in subroutine XSECT to indicate the species pairs for which cross section data have been supplied.

$ZM2(I)$

Quantity proportional to molecular weight of species $I$, eqs. (171)

4.60 Common /TRANS2/

$KKQ(L)$

Index of option to be used for step $L$ of cross section computations in input or precoded data; see Section 4.4 of Vol. II

$NNQ(L)$

Number of species pairs in $IIQ$, $JJQ$ arrays for which option $KKQ(L)$ is specified in input data

$ISEQ(K)$

Sequencing array specifying the order in which the defined steps of the transport cross section calculation are to be carried out

$NNKQ$

Number of steps specified for cross section computations

$I(J)$

Temporary array of species indices used by subroutine XSECT in preparing edited cross section data.
4.61 Common /TRANS3/

B(I,L) Intermediate variable in transport calculations; see KINT and KANDMU, Sections 3.27 and 3.28

BR(I,L) Reciprocal of B(I,L)

A(L) Intermediate variable in transport calculations; see KANDMU, Section 3.27

X(I) Mole fraction of species I in gas mixture

DH(I) Nondimensional internal specific heat \((W_i C_{pi}/N_0 k - 5/2)\) for species I in gas mixture

4.62 Common /TRANS4/

TL(I) Independent variable (temperature) for tabulated cross section data

ΩMEGAL(I) Value of first parameter determining cross sections at TL(I) for tabulated cross section data

ASTAR(I) Value of second parameter determining cross sections at TL(I) for tabulated cross section data

BSTAR(I) Value of third parameter determining cross sections at TL(I) for tabulated cross section data

4.63 Common /TRANS5/

N Number of species included in transport calculations

IELEC IELEC=0 indicates that electrons are not included in the calculations

IDL Index in the current gas model of first species used in Lewis number computation
Index in the current gas model of second species used in Lewis number computation

4.64 Common /TRANS7/

V(M) List of parameters for cross section computations in edited data

KQ(L) Option to be used for step L of cross section computations in edited data

NQ(L) Index of last species pair in IQ,JQ arrays to which step L of the edited cross section computations is to be applied

IQ(N) List of first indices for species pairs to which edited cross section data are to be applied

JQ(N) List of second indices for species pairs to which edited cross section data are to be applied

NKQ Number of steps in cross section computations for edited cross section data

4.65 Common /TRANS8/

NV(K) Number of parameters V required for option K of the cross section computations

N Number of species included in cross section computations

4.66 Common /TRANV/

VWi(K) For K = 1 to a maximum of 5, the input parameters for the i th step of the cross section calculations. The common block provides storage for i = 1 to 100
4.67 Common /TRPR9P/

VISC  Viscosity of mixture in poise

PRF  Frozen Prandtl number for mixture (dimensionless)

SIGMA  Electrical conductivity of mixture in mhos/cm

FLEWIS  Atom-molecule Lewis number of mixture (dimensionless)

4.68 Common /WMD/

TSF  Frozen stagnation temperature (°K)

4.69 Main Program

ATEST  Previous value of A in the Newton-Raphson iteration to determine the parameter \( \alpha \) in the density-area relation I(383)

ATH  Effective cross-sectional area of the flow at the throat, with allowance for the boundary layer displacement thickness (cm²)

CHTH  Nondimensional enthalpy \( hW_0/R_0T_0 \) at the throat

CMTH  Molecular weight of gas mixture at the throat

\( D \)  \( \left( \rho*/\rho_0 \right)^A (A+2)^{-2} \), where \( \rho* \) is the density at the sonic point in the equilibrium flow solution and \( A \) is the current estimate of the parameter \( \alpha \) in the density-area relation

DELT  Initial decrement of the nondimensional temperature for starting the perturbation calculation of the nonequilibrium flow

\( DL \)  \( \left( \rho*/\rho_0 \right)^A \left[ (A+2) \ln \left( \rho*/\rho_0 \right) + 1 \right] \)

ELTMIN  Case execution time in minutes
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ET</td>
<td>Elapsed time from beginning of execution (seconds)</td>
</tr>
<tr>
<td>ETO</td>
<td>Elapsed time from beginning of execution at the start of a case (seconds)</td>
</tr>
<tr>
<td>GJ(I)</td>
<td>Species concentrations in moles/g</td>
</tr>
<tr>
<td>GJA(I)</td>
<td>Species concentrations in the upstream reservoir, moles/g</td>
</tr>
<tr>
<td>GMF(J)</td>
<td>Mole fraction of the Jth species under equilibrium sonic conditions</td>
</tr>
<tr>
<td>HOP</td>
<td>Reservoir enthalpy (cal/g)</td>
</tr>
<tr>
<td>I</td>
<td>DΩ index; also, index, in the list of species for the current problem, of the Kth cold species</td>
</tr>
<tr>
<td>IRCO</td>
<td>Indicator; value 1 indicates that call to TRANSX was skipped in the previous case because tables of species thermal properties were generated</td>
</tr>
<tr>
<td>ISØLN</td>
<td>Indicator for type of flow solution:</td>
</tr>
<tr>
<td></td>
<td>ISØLN = 1 Frozen</td>
</tr>
<tr>
<td></td>
<td>ISØLN = 2 Equilibrium</td>
</tr>
<tr>
<td></td>
<td>ISØLN = 3 Nonequilibrium</td>
</tr>
<tr>
<td>IS3</td>
<td>Storage for input value of ISW3B; used to restore value after a case in which ISW3B was set to zero because NTRANS = .TRUE.</td>
</tr>
<tr>
<td>K</td>
<td>Cold species index</td>
</tr>
<tr>
<td>N</td>
<td>Iteration counter in Newton-Raphson solution for parameter α in area-density relation</td>
</tr>
<tr>
<td>NCASE</td>
<td>Counte− for cases in a run</td>
</tr>
<tr>
<td>NCØMPL</td>
<td>Number of cases successfully completed in a run</td>
</tr>
</tbody>
</table>
NERRØR  Number of failed cases in a run
NØZZLE  Nozzle profile index
NRCTØT  Total number of records written on tape 8 during a run
ØNE  1.0
PIV  3.14159
SUM1  \ln \left( \frac{\rho \ast}{\rho_0} \right)
SUM2  A+2
SUM3  \left( \frac{\rho \ast}{\rho_0} \right)^A
UTH  Nondimensional flow velocity (SU) at throat
VISCC  Viscosity in the reservoir (lbm/ft-sec)

4.70  Subroutine AESOLN

DELP(L)  Displacement thickness in cm, on Lth profile
DELSTP(L)  Displacement thickness at the throat in cm, on Lth profile
DENØM  One-fourth of the effective flow area at the throat for a channel (cm²)
DYDX  Slope \frac{dy}{dx} for the first profile in a channel
DZDX  Slope \frac{dz}{dx} for the second profile in a channel
FAC1  Effective ordinate of first profile in a channel (cm)
FAC2  Effective ordinate of second profile in a channel (cm)

-260-
Square root of the effective area ratio, $\sqrt{A_e}$

Axial coordinate in nozzle (cm)

Geometric ordinate of first profile in a channel (cm)

Geometric ordinate of second profile in a channel (cm)

4.7 Subroutine AGSØLN

Trial value of geometric area ratio

Effective area ratio

Constant term in equation relating the geometric and effective area ratios for a channel

Final solution for the geometric area ratio

Dummy argument in call to GEØMAR; represents X-derivative of geometric area ratio (not used)

Displacement thickness divided by RO, for boundary layer on Lth profile

Displacement thickness (cm) for boundary layer on Lth profile

Displacement thickness at the throat (cm) for boundary layer on Lth profile

Function which is 0 when the correct geometric area ratio has been found

F value in previous step of iteration

Iteration counter

DØ index over channel profiles

-261-
<table>
<thead>
<tr>
<th><strong>RNAME</strong></th>
<th>Routine name for call to DUMP routine</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SQRTA</strong></td>
<td>Square root of geometric area ratio</td>
</tr>
<tr>
<td><strong>UPDOWMN</strong></td>
<td>Indicator; 1.0 if downstream solution is desired by calling routine, -1.0 if upstream solution is desired</td>
</tr>
<tr>
<td><strong>X</strong></td>
<td>Axial coordinate in nozzle (cm)</td>
</tr>
<tr>
<td><strong>XL</strong></td>
<td>Temporary storage for previous value of X</td>
</tr>
<tr>
<td><strong>Xφ</strong></td>
<td>Value of X in previous iteration</td>
</tr>
<tr>
<td><strong>Y</strong></td>
<td>Geometric ordinate of first profile in a channel (cm)</td>
</tr>
<tr>
<td><strong>YOZ0</strong></td>
<td>$y_0z_0$, product of profile ordinates at the throat; one-fourth of the geometric throat area in a channel (cm²)</td>
</tr>
<tr>
<td><strong>Z</strong></td>
<td>Geometric ordinate of second profile in a channel (cm)</td>
</tr>
</tbody>
</table>

4.72 **Subroutine AXFIT**

| **AG** | Geometric area ratio |
| **UPDOWMN** | Indicator; -1.0 upstream of throat, 1.0 downstream |

4.73 **Subroutine AXSECT**

<p>| <strong>I(J)</strong> | List of species for which cross section data are wanted (common arr:.) |
| <strong>II</strong> | Index of first species in species pair |
| <strong>IV</strong> | Location of species II in the I array |
| <strong>JJ</strong> | Index of second species in species pair |
| <strong>JV</strong> | Location of species JJ in the I array |</p>
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>Running index for DΩ loops</td>
</tr>
<tr>
<td>LL</td>
<td>Step in cross section computations where data are to be added</td>
</tr>
<tr>
<td>LQ</td>
<td>Position of species pair in IQ,JQ list</td>
</tr>
<tr>
<td>LQ1</td>
<td>Position in IQ,JQ list where additional species pairs are to be added</td>
</tr>
<tr>
<td>N</td>
<td>Number of species pairs added to IQ,JQ list</td>
</tr>
<tr>
<td>NN</td>
<td>Number of species from I array to be considered in adding data</td>
</tr>
</tbody>
</table>

### 4.74 Subroutine BLAYER

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AE</td>
<td>Effective area ratio</td>
</tr>
<tr>
<td>AG</td>
<td>Geometric area ratio</td>
</tr>
<tr>
<td>AGJ</td>
<td>Factor proportional to $r^{2j}$ in numerical integration from $x_0$ to the first flow point</td>
</tr>
<tr>
<td>AG1</td>
<td>Geometric area ratio in numerical integration from $x_0$ to the first flow point</td>
</tr>
<tr>
<td>AJ</td>
<td>Factor proportional to $r^{2j}$</td>
</tr>
<tr>
<td>ALPR</td>
<td>Natural logarithm of the Prandtl number at the reference temperature</td>
</tr>
<tr>
<td>AM</td>
<td>Mach number</td>
</tr>
<tr>
<td>AML</td>
<td>Mach number in numerical integration from $x_0$ to the first flow point</td>
</tr>
<tr>
<td>AS</td>
<td>Sound speed (cm/sec)</td>
</tr>
<tr>
<td>AVCON</td>
<td>W/R0 parameter used in averaging the correlation parameter</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>AX</td>
<td>Intermediate quantity, eq. I(194d) in curvefit to shear parameter</td>
</tr>
<tr>
<td>B</td>
<td>Coefficient of n in linear approximation to $N(n,S_w)$, eq. I(164)</td>
</tr>
<tr>
<td>BLINTI</td>
<td>Value of integral $I$, eq. I(172), at the beginning of a step in the boundary layer solution</td>
</tr>
<tr>
<td>BM1</td>
<td>B-1</td>
</tr>
<tr>
<td>BX</td>
<td>Intermediate quantity, eq. I(194e), in curvefit to shear parameter</td>
</tr>
<tr>
<td>CC</td>
<td>$M_Ag$ in low Mach number region of integration from $\gamma_0$ to first flow point</td>
</tr>
<tr>
<td>CØSB(L)</td>
<td>Cosine of angle between tangent to Lth profile and the nozzle axis</td>
</tr>
<tr>
<td>C1</td>
<td>Intermediate quantity in quadratic solution for $dM/dx$</td>
</tr>
<tr>
<td>C2</td>
<td>Intermediate quantity in quadratic solution for $dM/dx$</td>
</tr>
<tr>
<td>DADX</td>
<td>$dA_g/dx$, derivative of geometric area ratio</td>
</tr>
<tr>
<td>DLM</td>
<td>$d \ln M/dx$, logarithmic derivative of Mach number</td>
</tr>
<tr>
<td>DMDX</td>
<td>$dM/dx$, derivative of Mach number</td>
</tr>
<tr>
<td>DUDX</td>
<td>Derivative of nondimensional velocity</td>
</tr>
<tr>
<td>DX</td>
<td>Axial distance from the previous flow point to the current one</td>
</tr>
<tr>
<td>DXI</td>
<td>Change in $\xi$ in the current step</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>DXØ</td>
<td>Size of last previous &quot;final&quot; step, apart from steps at model points</td>
</tr>
<tr>
<td>DXP</td>
<td>Distance from the point before the previous flow point to the previous flow point</td>
</tr>
<tr>
<td>DXSQ</td>
<td>( (DX)^2 )</td>
</tr>
<tr>
<td>DX2</td>
<td>( DX + DXP )</td>
</tr>
<tr>
<td>DYZDX(L)</td>
<td>Derivative of the radius of the ( L )th profile with respect to the axial coordinate</td>
</tr>
<tr>
<td>Ek1</td>
<td>Coefficient ( A ) in eq. I(164)</td>
</tr>
<tr>
<td>ENPRIM</td>
<td>( n' ), eq. I(172)</td>
</tr>
<tr>
<td>FINAL</td>
<td>Logical flag; &quot;false&quot; when BLAYER is called during an intermediate calculation of the non-equilibrium integration; &quot;true&quot; for the final calculation of conditions at each flow point</td>
</tr>
<tr>
<td>F0</td>
<td>Intermediate quantity in curvefit to shear parameter, eq. I(194)</td>
</tr>
<tr>
<td>F2</td>
<td>Intermediate quantity in curvefit to shear parameter</td>
</tr>
<tr>
<td>F3</td>
<td>Intermediate quantity in curvefit to shear parameter</td>
</tr>
<tr>
<td>F4</td>
<td>Intermediate quantity in curvefit to shear parameter</td>
</tr>
<tr>
<td>GAMMAE</td>
<td>Effective specific heat ratio, ( \gamma_0 = a^2 W/RT ), used in calculation of effective hypersonic parameter</td>
</tr>
<tr>
<td>GJ(I)</td>
<td>Species concentrations (moles/g)</td>
</tr>
<tr>
<td>H</td>
<td>Nondimensional enthalpy, ( hW_0/R_0T_0 )</td>
</tr>
</tbody>
</table>

-265-
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HC0N</td>
<td>( R_0^2/W_0 ), factor for converting nondimensional enthalpy into cal/g</td>
</tr>
<tr>
<td>HE</td>
<td>Free-stream specific enthalpy (cal/g)</td>
</tr>
<tr>
<td>HF</td>
<td>Form factor, ( c^*/\theta )</td>
</tr>
<tr>
<td>HFP1</td>
<td>HF + 1</td>
</tr>
<tr>
<td>HREF</td>
<td>Reference enthalpy</td>
</tr>
<tr>
<td>HTR</td>
<td>Incompressible form factor, ( H_{inc} )</td>
</tr>
<tr>
<td>HTRP1</td>
<td>HTR + 1</td>
</tr>
<tr>
<td>HW</td>
<td>Enthalpy of the gas at the wall temperature</td>
</tr>
<tr>
<td>HG0</td>
<td>Specific stagnation enthalpy (cal/g)</td>
</tr>
<tr>
<td>I</td>
<td>D( \phi ) index</td>
</tr>
<tr>
<td>ITYPE</td>
<td>Index for type of nozzle geometry</td>
</tr>
<tr>
<td>J</td>
<td>D( \phi ) index</td>
</tr>
<tr>
<td>K</td>
<td>D( \phi ) index over the cold species</td>
</tr>
<tr>
<td>L</td>
<td>D( \phi ) index running over the nozzle or channel profiles</td>
</tr>
<tr>
<td>LPR</td>
<td>Index, equal to LPRIME(L)</td>
</tr>
<tr>
<td>LPRIME(L)</td>
<td>Equal to 2 for ( L = 1 ), equal to 1 for ( L = 2 )</td>
</tr>
<tr>
<td>( \phi )LDMF</td>
<td>Logical flag; &quot;true&quot; if no mole fraction has changed by 0.01 since the last flow point at which transport properties were calculated</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$\Omega$</td>
<td>$\omega$, viscosity-temperature exponent in eq. I (191)</td>
</tr>
<tr>
<td>$\Omega_1$</td>
<td>$\text{Max} ,(1-\omega,0)$</td>
</tr>
<tr>
<td>$\Omega_n$</td>
<td>1.</td>
</tr>
<tr>
<td>$\Omega_{\text{RDIN}}(L)$</td>
<td>$\bar{x}$ for the boundary layer on the $L$th profile, eq. I(171)</td>
</tr>
<tr>
<td>$\Omega_{\text{RD1}}$</td>
<td>A value of $\bar{x}$ used in the numerical integration from $x_0$ to the first flow point</td>
</tr>
<tr>
<td>$\Omega_{\text{RD2}}$</td>
<td>See $\Omega_{\text{RD1}}$</td>
</tr>
<tr>
<td>$P$</td>
<td>Nondimensional pressure, $p/p_0$</td>
</tr>
<tr>
<td>$\text{PRPWR}$</td>
<td>$(\text{PRREF})^{0.56}$</td>
</tr>
<tr>
<td>$\text{PRW}$</td>
<td>Prandtl number at the wall temperature</td>
</tr>
<tr>
<td>$R$</td>
<td>Ratio of local nozzle radius to throat radius for axisymmetric nozzles</td>
</tr>
<tr>
<td>$\text{RA}$</td>
<td>Intermediate quantity in curvefit to Reynolds analogy factor</td>
</tr>
<tr>
<td>$\text{RAF}$</td>
<td>Reynolds analogy factor $R_A$ in eq. I(195)</td>
</tr>
<tr>
<td>$\text{RB}$</td>
<td>Intermediate quantity in curvefit to Reynolds analogy factor</td>
</tr>
<tr>
<td>$\text{RER}$</td>
<td>Reynolds number based on $R_0$</td>
</tr>
<tr>
<td>$\text{RH}$</td>
<td>Nondimensional density</td>
</tr>
<tr>
<td>$\text{RMR}$</td>
<td>$\rho \mu$ ratio, $\rho_w \mu_w/\rho_e \mu_e$</td>
</tr>
<tr>
<td>$\text{RWMW}$</td>
<td>$\rho_w$</td>
</tr>
<tr>
<td>$\text{R1}$</td>
<td>Radius of a profile in the numerical integration from $x_0$ to the first flow point</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>R2</td>
<td>See R1</td>
</tr>
<tr>
<td>SIGH</td>
<td>Hypersonic parameter $\sigma$</td>
</tr>
<tr>
<td>SQT</td>
<td>Intermediate quantity in transverse curvature correction, eq. 1(187)</td>
</tr>
<tr>
<td>SWFAC</td>
<td>$(S_w + 1)[7 + 3.4(S_w + 1)]$</td>
</tr>
<tr>
<td>SWFAC2</td>
<td>$1 + 4.5(S_w + 1)^{0.9}$</td>
</tr>
<tr>
<td>SWP1</td>
<td>$S_w + 1$</td>
</tr>
<tr>
<td>T</td>
<td>Nondimensional temperature, $T/T_0$</td>
</tr>
<tr>
<td>TE</td>
<td>Free stream temperature, $T$</td>
</tr>
<tr>
<td>TEP</td>
<td>Free stream temperature at which transport property calculations were last done</td>
</tr>
<tr>
<td>TETAL</td>
<td>Momentum thickness based on eq. 1(185), divided by $\rho_0$</td>
</tr>
<tr>
<td>TR</td>
<td>$T^*/T_w$, ratio of reference temperature to wall temperature</td>
</tr>
<tr>
<td>TRCALC</td>
<td>Logical flag, .TRUE. if transport properties were calculated at the current flow point</td>
</tr>
<tr>
<td>TREF</td>
<td>Reference temperature (K)</td>
</tr>
<tr>
<td>TREFP</td>
<td>Reference temperature at which transport properties were last calculated</td>
</tr>
<tr>
<td>TRP(I)</td>
<td>Array equivalenced to common block /TRPR@P/, to permit processing these data in a DO loop</td>
</tr>
<tr>
<td>TRPSV(I)</td>
<td>Array used to save the transport properties at the free stream temperature</td>
</tr>
<tr>
<td>U</td>
<td>Nondimensional velocity, $u/\mu_s$ (see US in common block /SS/)</td>
</tr>
</tbody>
</table>
UPRIME  Free stream velocity, u (cm/sec)
VISCR    Viscosity at the reference temperature (poise)
VISCW    Viscosity at the wall temperature (poise)
VR       VISCR/VISCW
WD       Weight factor w in calculation of d δ*/dx; eq. I(419)
WW       $e^{-\Delta x/a}$ in eq. I(217); weighting factor for \( \bar{\eta}_p \) in the calculation of the averaged correlation parameter, \( \bar{n} \)
X        Axial coordinate (cm)
XII      Previous value of XI
XL       Shear parameter \( \ell \), eqs. I(194)
XP       Previous value of x (cm)
XPXP     X + XP
XSAVE(I) Temporary storage for species mole fractions SAVEC(I)
XSNV     min (\( \bar{n} \), 0); used in place of \( \bar{n} \) in calculation of HTR to ensure the use of a nonpositive value
XTEST(J) Mole fraction of the Jth species used in the last previous calculation of transport properties
XX       Axial coordinate variable in numerical integration from \( x_0 \)
YZ(L)    Radius of the Lth profile (cm)
ZERO     0.
4.75 Subroutine BICALL

FINAL Logical subroutine argument; .FALSE. when
BLAYER is called during an intermediate cal-
culation of the nonequilibrium integration,
.TURE. for the final calculation of condi-
tions at the end of each successful integra-
tion step

GJ(J) Species concentrations in moles/g

J DØ index over the species

4.76 Subroutine BXSECT

I First index of species pair in V array

J Second index of species pair in V array

K Value of KQ for step L of cross section compu-
tations

L Index of step in cross section computations
which is currently being considered

M Number of sets of constant factors to be used
in computations, for option KQ = 14 only

MADD Index used in locating species pairs in V
array

MM MM + 1 and MM +2 are the locations of the cur-
rent species pair in the V array

MV Index of last parameter in V array for step
L-1 of the computations

4.77 Subroutine CØMM

ALGJ(J) ln γ_j, natural logarithm of the jth species
concentration
CLKF \( \ln k_f \), natural logarithm of the forward rate constant

CLKR \( \ln k_r \), natural logarithm of the reverse rate constant

CLTBF \( \ln (\Sigma \rho \gamma_j) \), where the sum is over the third-body species in a reaction

CLZ2 \( \ln (Z2) \)

CLZ3 \( \ln (Z3) \)

EF \( \epsilon_{fi} \) in eq. I(32lc)

ER \( \epsilon_{ri} \) in eq. I(32lc)

GJ(J) Concentration \( (\gamma_j) \) of the Jth species (mole/g)

I D\( \phi \) index over the reactions

ICHG Equal to 1 for a neutral species, equal to 2 for a charged (ionic) species

IC\( \Omega \)UNT Counter for number of times C\( \Omega \)MM has been called in a run

ICCYCLE Interval in IC\( \Omega \)UNT at which ISW5B is set equal to 1 to provide diagnostic dumps in subroutine C\( \Omega \)MM, EXACT, RNKT, and PRTA

IPAI Index of atomic species produced in current reaction for KTF(I) = 4

J D\( \phi \) index over the species

KF Indicator for temperature dependence of forward reaction rate constant; for KF = 1, \( k_f \) is a function of \( T \); for KF = 2, it is a function of \( T_c \). For other values of KF, see discussion of KTF input in Sec. 2.3 of Vol. II.
KR  Indicator for temperature dependence of reverse reaction rate constant; for KR = 1, $k_r$ is a function of $T$; for KR = 2, it is a function of $T_e$

LQGC  Logical indicator for method of calculation of PICH(1); for LQGC = .FALSE., PICH(1) is computed as $P_i \Gamma_i$; for LQGC = .TRUE., it is calculated as $-\exp[\ln P_i + \ln (1 - \Gamma_i)]$ to avoid possible underflow of the exponential function in $P_i = \exp (\ln P_i)$

ONE  1.

QEILAS  Energy transfer to the electron gas as a result of elastic collisions (cal/cm$^3$ sec)

QF  $q_{fi}$ in eq. I(321b)

QR  $q_{ri}$ in eq. I(321b)

R  Effective nozzle radius for loss of resonance radiation

RCALC  Logical indicator; if .TRUE., R has already been computed for current flow point

RKF  $k_f$, forward rate constant

RKR  $k_r$, reverse rate constant

SUMF  $\sum_{j=1}^{n} \gamma_{ij} (\ln \gamma_i + \ln \gamma_j)$

SUMR  $\sum_{j=1}^{n} \gamma'_{ij} (\ln \rho + \ln \gamma_j)$

TAU  Optical thickness based on R for resonance radiation

TEP  Electron temperature, $T_e$ (°K)
Electron-ion elastic collision frequency ($\nu_{ei}$) divided by the ionic species concentration $\gamma_j$

$$\nu_{ei} = \frac{(8/3)\sqrt{\pi/m_e} \ e^4 N_0/(2k)^{3/2}}{N_j}$$

where $m_e$ is the electron mass (g), $e$ is the electronic charge (esu), $N_0$ is Avogadro's constant, and $k$ is Boltzmann's constant.

Electron-neutral collision frequency ($\nu_{ej}$) divided by the neutral species concentration ($\gamma_j$)

$$\nu_{ej} = \frac{k^3/(\pi e^6)}{\sqrt{8k/(\pi m_e)} N_0}$$

Radius of first channel profile, cm

Radius of second channel profile, cm

=$\ln u$, natural logarithm of flow velocity in cm/sec

$$R, T'$$, where $T' = T$ for $KR = 1$ and $T' = T_e$ for $KR = 2$ (cm$^3$ atm/mole)

$R_0 T_0$ (cm$^3$ atm/mole)

$\rho R_0 T'$ (see Z2)
4.78 Subroutine CXSECT

I(J) New index of species for which old index is J

J Second index of species pair

L First index of species pair

M Temporary variable

4.79 Subroutine DERIVS

CPJE Temporary storage for specific heat of the electrons

CTSAVE Temporary storage for the nondimensional gas temperature T/T₀

DELI1(L) \( S_{i1}^* \) for the Lth profile (eq. I(424));

DELI2(L) \( S_{i2}^* \) for the Lth profile (eq. I(424));

DELØ1(L) \( S_{o1}^* \) for the Lth profile (eq. I(424));

DELØ2(L) \( S_{o2}^* \) for the Lth profile (eq. I(424));

DEN Denominator in eq. I(424)

HJE Temporary storage for nondimensional enthalpy of the electron gas (Hₑ/R₀T₀)

I DØ index over species

ITRY Counter for iteration to obtain self-consistent solution for the boundary layer displacement thickness (Section 7.6 of Vol. I)

K DØ index over profiles

SENTE Temporary storage for nondimensional entropy of the electron gas (Sₑ/₀/R₀)
Matrix initially containing the coefficients and constants for a system of linear equations to be solved:

\[ a_{11}x_1 + a_{12}x_2 + \cdots + a_{1i}x_i = b_1 \]
\[ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2i}x_i = b_2 \]
\[ \vdots \]
\[ a_{ii}x_1 + a_{i2}x_2 + \cdots + a_{ii}x_i = b_i \]

The coefficients \( a_{jk} \) are stored in the locations \( A(j,k) \), for \( j=1 \) to \( i \) and \( k=1 \) to \( i \). The constant terms \( b_j \) are stored in \( A(j,i+1) \), for \( j=1 \) to \( i \). The solution values \( x_k \) are written into the locations \( A(k,i+1) \) for \( k=1 \) to \( i \), that is, the locations originally occupied by the \( b_j \).

**I**
Number of equations in the system to be solved

**J**
DØ index running from 1 to \( I \)

**K**
DØ index running from 1 to \( I \)

**KS**
Error indicator; nonzero value indicates singular matrix of coefficients

**L**
Counter for matrix elements during rewriting of the matrix in close-packed form

**LM**
First index for close-packed form of matrix

**LN**
Number of columns already filled in close-packed matrix

**LNL**
Column index (second index) in close-packed matrix

**M**
\( I + 1 \)
N
First dimension of A in the dimension statement in the calling routine

N1
N + 1

RNAME
Routine name for calling DUMP routine

4.81 Subroutine DUMP

RNAME
Name of routine from which DUMP was called

4.82 Subroutine DUMP

No local variables.

4.83 Subroutine ELCOND

I
Species index

SIGMA
Electrical conductivity in mhos/cm

SSIG
\((e/k)^2/\sqrt{T}\)

4.84 Subroutine ELTIME

CALLED
Logical indicator, initially .FALSE., set to .TRUE. when ELTIME is first called

DET
Elapsed time since last printed time (seconds)

ET
Elapsed time from beginning of execution (seconds)

ETM
Elapsed time from beginning of execution (minutes)

ETØ
ET argument value passed from calling routine

ETP
Previous value of ETØ

I
Elapsed time from start of execution in milliseconds
Control for printing the time; zero value suppressed the print

4.85 Subroutine EPART

Energy gained by the electron gas in the forward direction of a reaction (cal/mole)

Negative of the energy gained by the electron gas in the reverse reaction (cal/mole)

Reaction energy to be partitioned between radiative losses and heating of the electron gas, $E_0$ (cal/mole)

Reaction index in the list of reactions for the current gas model

Indicator (ITR) of the rule for partitioning the reaction energy in the current reaction

Energy lost by radiation in the forward reaction (cal/mole)

Negative of energy lost by radiation in the reverse reaction (cal/mole)

Universal gas constant, cal/mole-°K

Electron temperature (°K)

$3R_0/2$, where $R_0$ is the universal gas constant in cal/mole-°K

4.86 Subroutine EOCALC

Variable used to save and restore the initial value of the nondimensional temperature CT

DΦ index

I + ISC
If ISC
Iteration counter for Newton-Raphson solution of equations I(233)

P
Pressure (atm)

RNAME Routine name for calling DUMP routine

T Temperature (°K)

ZA Newton-Raphson correction factor \(1 + h_n \gamma\) to mole fractions, eq. I(235)

4.87 subroutine **EXACT**

AAA(L,M) Array of coefficients and constants for system of linear equations

DGJ(J) \(\frac{d \gamma_j}{dx}\)

ECØEF \(1/(\rho u R_0T_0)\)

GJ(J) \(\gamma_j\), species concentrations, moles/g

I DØ index

J DØ index over species

K I - ISC

NE Number of equations to be solved for rates of change of dependent variables during the non-equilibrium integration

SUMTDG \(\sum_{j=1}^{N} \frac{T}{T_0} \frac{d \gamma_j}{dx}\) for chemical nonequilibrium models; \(\sum_{j=2}^{N} \frac{T}{T_0} \frac{d \gamma_j}{dx}\) for electronic nonequilibrium models
\[ s_j = -1 + \nu_{i-c} + \frac{H_i}{R_0^T} - \sum_{j=1}^{c} \nu_{i-c,j} \frac{H_j}{R_0^T} \]

\[ l = \left(\frac{u}{u_s}\right)^2 \frac{1}{W_0} \]

**4.88 Function EXP**

**EXP** Value of function

**X** Subroutine argument

**4.89 Subroutine FINDX**

**A** Geometric area ratio (from calling routine)

**AR** Geometric area ratio based on a trial value of X

**DADX** Derivative of geometric area ratio with respect to X (cm\(^{-1}\))

**DZDX(I)** Slope of Ith profile in channel

**ERA** Error in geometric area ratio AR at current value of X

**ERRL** Smallest negative error in AR for any previous trial value of X

**ERRU** Smallest positive error in AR for any previous trial value of X

**IENTRY** Indicator of entry point; 1 for FINDX, 2 for FINDXIC

**MBL** Index (1 or 2) of the profile whose ordinate varies most rapidly downstream of the throat in a channel

**N** Iteration counter
RNAEIE

Routine name for call to DUMP routine

UPDOWN

Indicator; -1.0 if upstream solution is desired, 1.0 if downstream solution is desired

UPPER

Logical indicator; .TRUE. if a positive ERA has been found at some trial value of X

V

Trial value of |X| (cm)

VL

Trial value of |X| corresponding to error ERRL

VØ

Trial value of |X| in previous iteration step

VU

Trial value of |X| corresponding to error ERRU

X

Final solution for X (cm)

Z(I)

Ordinate of the Ith profile in a channel

4.90 Subroutine FROZEO

AR

Common variable, used here for the geometric area ratio

AS

Sound speed (cm/sec)

BLVAR(I)

Array equivalenced to the contents of common block /BLOUT/

BLVARø(I)

BLVAR values at preceding flow point

BLVARS(I)

BLVAR values at the current regular flow point; used in interpolating at a model point

CLNMA

Logarithm of the molecular weight in the reservoir

CTSAVE

Nondimensional temperature at the regular flow point following a model point
<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTSTAR</td>
<td>The nondimensional temperature (CT) value at the flow point preceding the throat, in solutions including the boundary layer</td>
</tr>
<tr>
<td>CT1</td>
<td>Second most recent trial value of CT in the iteration to find the conditions at a model point</td>
</tr>
<tr>
<td>CT2</td>
<td>Most recent trial value of CT in iteration to find the conditions at a model point</td>
</tr>
<tr>
<td>CXSAVE</td>
<td>X coordinate of the regular flow point following a model point</td>
</tr>
<tr>
<td>DELBLØ(I)</td>
<td>For I = 1, 2, the nondimensional boundary layer displacement thickness for the Ith profile at the preceding flow point</td>
</tr>
<tr>
<td>DELBLP(I)</td>
<td>For I = 1, 2, the nondimensional boundary layer displacement thickness for the Ith profile at the point before the preceding flow point</td>
</tr>
<tr>
<td>DELT</td>
<td>Decrement in nondimensional temperature CT in calculation of frozen throat conditions</td>
</tr>
<tr>
<td>DELTSV</td>
<td>Variable used to save the input value of DELT1</td>
</tr>
<tr>
<td>DELTV</td>
<td>Current decrement in nondimensional temperature CT</td>
</tr>
<tr>
<td>ERRX</td>
<td>Position error in iteration to determine the conditions at a model point</td>
</tr>
<tr>
<td>FLAG</td>
<td>Indicator for upstream and downstream regions, for use by the geometry subroutines. Negative value indicates an upstream position, positive value a downstream position</td>
</tr>
</tbody>
</table>
Values of nondimensional mass flux in iteration to determine the frozen throat conditions.

Species concentrations, mole/g

DO index

Indicator for iteration to take the boundary layer displacement thickness into account in the determination of reservoir conditions. IPASS = 1 causes a return to the main program after the throat has been reached.

Counter for number of times solution has been restarted with a lower value of W to suppress instability due to coupling of the boundary layer and the inviscid solution.

Indicator for type of flow solution:
- ISOLN = 1 Frozen solution
- ISOLN = 2 Equilibrium solution

Indicator for region of flow solution:
- ISOMIC = -1 Upstream region
- ISOMIC = 0 Sonic point
- ISOMIC = 1 Downstream region

Counter for number of flow points beyond the throat.

Index of the next specified test section diameter (TSDIAM)

DO index over gas species

DO index

Indicator for last point in flow solution (at CX = CXMAX)

DO index over channel profiles

Logical indicator; if .TRUE., a model point is being calculated

DO index
NMPI  Counter for number of iterations to determine the conditions at a model point

ØNE  1.0

SØLIP  Logical variable used to indicate that conditions at a model point have just been determined

SØLN(I,J)  Hollerith array

SWITCH  Logical indicator; if .TRUE., FRØZEQ has begun taking the displacement thickness into account in calculations of the flow-point position

\[ S_1 = -\frac{1}{W_0} \sum_{j=1}^{n} X_{j0} \ln X_{j0} \]

\[ S_2 = -(\ln p_0)/W_0 \]

TSTEPCE  Logical indicator for special control of temperature decrement far downstream, to prevent distance of successive flow points from becoming too large

T1,T2,T3  Nondimensional temperature values in iteration to determine frozen throat conditions

XFRAC  Distance from preceding flow point to model point, divided by distance from preceding flow point to current point

XI LAST  Position of preceding flow point (cm)

XMØDEL  Model-point position (cm)

XMØDL1  Position of the next model point in the geometric sequence of model points (cm)

XØ  Position of the preceding flow point (cm)

XP  Position of the point before the preceding flow point (cm)
Position of the flow point corresponding to the second most recent trial value of CT in the iteration to find the conditions at a model point (cm)

Position of the flow point corresponding to the most recent trial value of CT in the iteration to find the conditions at the model point (cm)

F1 - F2 + F3 - F2

4.91 Subroutine GEOM

Effective area ratio corresponding to the current trial value of \( \rho \), based on the area ratio-density relation, eq. I(383)

Square of the effective area ratio

\( 2C(A_e-1)/ \alpha \)

\( \rho_0 \frac{dA_e}{d\rho} \), derivative of the effective area ratio with respect to nondimensional density

Correction in nondimensional density

Absolute value of the difference between \( \rho/\rho_0 \) and the nondimensional density at the throat, based on approximate solution near throat

Displacement thickness at the throat (cm)

\( \left[ \frac{d^2(ln A_e)}{dx^2} \right] \), the second derivative of the logarithm of the effective area ratio at the throat

\( \omega \) index over the profiles in a channel

Iteration counter

Number of upstream sections in the curvefit to a profile
\( \phi \text{PHA} \) 
1 + \( \alpha/2 \), where \( \alpha \) is the exponent in the density-area ratio relation \( I(383) \)

\( \text{RNAME} \) 
Routine name for call to DUMP

\( \text{SGN} \) 
-1 if \( dA_\rho/dx \) is positive, +1 if \( dA_\rho/dx \) is negative

\( \text{Y} \) 
Consistency check on \( x \) and \( \rho \) values; if positive, \( x \) and \( \rho \) data are inconsistent

\( \text{Z} \) 
\( (\rho/\rho_0)^\alpha \)

4.92 Subroutine GE\( \phi \)MAR

\( \text{ARATI} \phi \) 
Geometric area ratio

\( \text{DAB} \) 
Difference between \( X \) and the \( x \)-coordinate of the center of a circular profile section

\( \text{DERIVA} \) 
\( dA_\rho/dx \), derivative of geometric area ratio (cm\(^{-1}\))

\( \text{DRV} (I) \) 
Slope (derivative) of ordinate for Ith profile

\( \text{DYDX} \) 
Slope of ordinate for first profile, after rescaling area

\( \text{DZDX} \) 
Slope of ordinate for second profile, after rescaling area

\( I \) 
Index of two profiles

\( \text{IENTRY} \) 
Index of entry point: 1 - GE\( \phi \)MAR; 2 - GMAR; 3 - GMAR2; 4 - GMAR3

\( \text{ISJ} \) 
Shape index for a profile section
\( J \)  
Index for the sections in a profile

\( K \)  
Index for the profile section in which \( X \) lies

\( \text{NSM1} \)  
Number of sections in a profile minus 1

\( Q \)  
Difference in ordinate between a point on a circular profile section and the circle center (cm)

\( \text{RAD(I)} \)  
Ordinate of the Ith profile (cm)

\( \text{RAT} \)  
Ratio of a profile ordinate to the ordinate at the throat

\( \text{SN(I)} \)  
\(-1.0 \) for \( I = 1 \), \(+1.0 \) for \( I = 2 \)

\( \text{SQTRSA} \)  
Square root of area rescaling factor RSA

\( S1(I) \)  
Unrescaled geometric area ratio; or ratio of ordinate to ordinate at the throat for Ith profile

\( S2(I) \)  
Derivative of geometric area ratio; or ratio of derivative of ordinate to ordinate at the throat for Ith profile (cm\(^{-1}\))

\( X \)  
Axial coordinate in nozzle (cm)

\( Y \)  
Ordinate of first profile (cm)

\( Z \)  
Ordinate of second profile (cm)

4.93 \textbf{Subroutine INGAS}

\( \text{CAPQ(K)} \)  
\( Q_k \), number of gram-atoms of the kth chemical element per mole of the cold gas

\( \text{CCI(I)} \)  
Intermediate variable in the calculation of the molecular weights of species from atomic weights of elements
I  Species index
II  Index running from c + 1 to n
J  Element index
K  Index running from 1 to c
L  Index running from c + 1 to n
SUM Dummy variable used for accumulating summations

4.94 Subroutine INIT
I  Species index in the current gas model
II  Species index in the master list of species
J  Index of additional vibrational modes in tri-atomic molecules
K  Number of electronic levels for a species
L  Index of the electronic levels for a species
SPRP(M,II) Array containing species property data in common /SPEC/
T1  Reservoir temperature $T_0(\text{OK})$
T2  $T_0^2$
T3  $T_0^3$
T4  $T_0^4$

4.95 Subroutine KANDMU
I  Species index
II  I-1
Species index

Index of property being calculated; \( L = 1 \) for translational thermal conductivity, \( L = 2 \) for viscosity

Weighting factor used in computing \( \tilde{a}(\alpha) \) from eq. I(86)

Variable used to accumulate numerator in eq. I(86)

Temporary variable used in computing denominator in eq. I(86)

Temporary variable used in computing numerator in eq. I(86)

Temporary variable used for accumulating denominator in eq. I(86) and numerator in eq. I(86)

\[ ZK(L) \cdot \sqrt{T} \] is the translational thermal conductivity in milliwatts/cm\(^{-1}\)K, \( ZK(2) \cdot \sqrt{T} \) is the viscosity in millipoise

4.96 Subroutine KINT

Species index

I-1

Summation index for sum over species pairs with \( J < I \)

Index of property being computed (see discussion of KINT in Sec. 3.28)

Species index

Summation index for sum over species pairs with \( NJ > NI \)
N1  N + 1
ZKINT  \( ZKINT \sqrt{T} \) is the internal thermal conductivity in milliwatt/cm\(^{-\circ K}\)

4.97 Subroutine LIST

CAPQ(I) \( Q_k \), number of gram-atoms of the \( k \)th chemical element per mole of the cold gas

C\( OM1(I,J) \) Hollerith array for output (see DATA statement)

C\( OM2(I,J) \) Hollerith array for output (see DATA statement)

CXMAXI Distance beyond throat at which case will be terminated (inches)

I  D\( \Phi \) index

J  D\( \Phi \) index

K  D\( \Phi \) index

KDQ1 KDIM + 1

NWXI Number of specified distances WXI from the leading edge of a wedge model

SBJ(J) \( b_j \), eqs. I(51)

SHJAP(J) Formation enthalpy of the \( J \)th species, cal/mole

THEVP(J) Characteristic vibrational temperature for the \( J \)th species, \(^{\circ K}\)

W\( \Omega RD(I,J) \) Hollerith array for output (see DATA statement)
4.98 **Subroutine MATINV**

A  Original matrix; replaced by inverse

AA  Temporary storage used in interchanging columns

B  Temporary storage used in interchanging rows

BB  Temporary storage used in reduction of off-diagonal elements

BIG  Pivotal element in a column (largest absolute magnitude)

I  DØ index

IV(K)  Index array for recording interchanges of rows

J  Temporary storage used in interchanging IV elements; also DØ index

JA  Index of current row; reset to index of next row

JB  Index of preceding row

JJ  Index of current row

K  First index of pivotal element in a column

KL  K + 1

L  DØ index

M  Equal to MM

MM  Size of matrix to be inverted

ML  M - 1

NN  First dimension of A in dimension statement of calling routine
Routine name for calling DUMP routine

4.99 Subroutine MODEL

ALPHA Ratio of the current estimates of the pressure $p_2$ and the density $\rho_2$ behind the shock

CATM Conversion factor from atm to dyne/cm$^2$

CKSL Parameter characterizing low density effects on the stagnation-point heat flux, $E K^2$ (per foot)

CML Mean molecular weight of the gas ahead of the shock, $g$/mole

CTSAVE Temporary storage for nondimensional free stream temperature CT

DET Computer time used in model calculations, seconds

DISC Discriminant in quadratic solution for the pressure behind the equilibrium normal shock

EIN(I) Exponent of Lewis number in Fay-Riddell heat transfer correlation (equilibrium boundary layer. $I = 1$, frozen boundary layer $I = 2$)

EMF Temporary storage for electron mole fraction at stagnation point, to permit output of electron concentration

EPS Density ratio $\varepsilon = \rho_1 / \rho_2$ across normal shock

EPSOLD Previous estimate of $\varepsilon$

EQUALS Hollerith equality sign used in output

ERRHI Most recent positive error in the nondimensional enthalpy in iterative solutions for conditions behind the normal shock and at the inviscid stagnation point
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERIØ</td>
<td>Most recent negative error in the nondimensional enthalpy</td>
</tr>
<tr>
<td>ERZ</td>
<td>Current error in the nondimensional enthalpy</td>
</tr>
<tr>
<td>ET</td>
<td>Elapsed time since the beginning of the run, seconds</td>
</tr>
<tr>
<td>ETØ</td>
<td>Elapsed time since the beginning of the run when MODEL is entered, seconds</td>
</tr>
<tr>
<td>GJ(I)</td>
<td>Species concentrations, moles/g</td>
</tr>
<tr>
<td>HCF</td>
<td>Enthalpy of formation of the gas mixture at the stagnation point</td>
</tr>
<tr>
<td>HCØN</td>
<td>Conversion factor $(RT_0/W_0)$ from nondimensional enthalpy to enthalpy in cal/g</td>
</tr>
<tr>
<td>HD</td>
<td>Nondimensional kinetic energy per unit mass of the gas ahead of the shock, $(u/u_s)^2/2$</td>
</tr>
<tr>
<td>HE</td>
<td>Free-stream static enthalpy (cal/g) ; used in flat-plate calculation</td>
</tr>
<tr>
<td>HI</td>
<td>Logical variable; if .TRUE., a positive enthalpy error has already been obtained in the iteration</td>
</tr>
<tr>
<td>HR</td>
<td>Recovery enthalpy (cal/g) in flat-plate calculation</td>
</tr>
<tr>
<td>HRATIPØ</td>
<td>Ratio of the dissociation enthalpy to $(h_e-h_w)$</td>
</tr>
<tr>
<td>HSTAG</td>
<td>Enthalpy at the inviscid stagnation point, cal/g</td>
</tr>
<tr>
<td>HSTAR</td>
<td>Reference enthalpy (cal/g) in flat-plate calculation</td>
</tr>
<tr>
<td>HW</td>
<td>Gas enthalpy at the wall temperature of the stagnation point model, cal/g</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>H2</td>
<td>Nondimensional enthalpy behind the normal shock</td>
</tr>
<tr>
<td>I</td>
<td>DØ index</td>
</tr>
<tr>
<td>ICOUNT</td>
<td>Iteration counter</td>
</tr>
<tr>
<td>ISØLN</td>
<td>Index specifying the two types of normal-shock solution (equilibrium ISØLN = 1, frozen ISØLN = 2)</td>
</tr>
<tr>
<td>ISS5P1</td>
<td>Number of lines of output required for the mole fractions at the inviscid stagnation point</td>
</tr>
<tr>
<td>J</td>
<td>Index of chemical elements</td>
</tr>
<tr>
<td>K</td>
<td>DØ index</td>
</tr>
<tr>
<td>L</td>
<td>Model-shape index</td>
</tr>
<tr>
<td>LIMSCØ(I)</td>
<td>For I = 1, 2, stored values of LMSC for the equilibrium and frozen normal shock solutions</td>
</tr>
<tr>
<td>LIM1</td>
<td>Lower limit of implied DØ loop for output of mole fractions</td>
</tr>
<tr>
<td>LIM2</td>
<td>Upper limit of implied DØ loop for output of mole fractions</td>
</tr>
<tr>
<td>LMSC</td>
<td>Number of data items from the SCØUT array to be printed for the current normal shock solution</td>
</tr>
<tr>
<td>LØ</td>
<td>Logical indicator; if .TRUE., a negative enthalpy error has already been obtained in the iteration</td>
</tr>
<tr>
<td>N</td>
<td>DØ index</td>
</tr>
<tr>
<td>NEL</td>
<td>Number of chemical elements in a species</td>
</tr>
</tbody>
</table>
PC \[4 \rho_1^2 u_1^2, \text{ atm-}g/\text{cm}^3\]

PRW Prandtl number at the model wall temperature

PS Stagnation pressure, atm

PT \[p_1 + \rho_1 u_1^2, \text{ atm}\]

P1 Pressure ahead of the shock, atm

P2 Pressure behind the normal shock, atm

QFRE(L) Stagnation point heat flux times the square root of the nose radius for an equilibrium boundary layer (hemispherical model \(L = 1\), flat-faced model \(L = 2\)), Btu/ft\(^{3/2}\)-sec

QFRF(L) Stagnation point heat flux times the square root of the nose radius for a frozen boundary layer (hemispherical model \(L = 1\), flat-faced model \(L = 2\)), Btu/ft\(^{3/2}\)-sec

QF Factor used in calculating QFRE and QFRF

QS Stagnation point heat flux times the square root of the nose radius based on the SRI approximation, Btu/ft\(^{3/2}\)-sec

QSRIT Stagnation point heat flux times the square root of the nose radius based on the SRI approximation, Btu/ft\(^{3/2}\)-sec

QUANT(I) Hollerith names of output quantities in SCOUT array

RH\(\text{O}\)STR Density at reference temperature in flat-plate calculation

R1 Density ahead of the shock, g/cm\(^3\)

R1U12 \[\rho_1 u_1^2, \text{ atm}\]

R2 Density behind the shock, g/cm\(^3\)

SCOUT(I) Array used for output of model conditions
SHV

Array equivalenced to common block /STAG/ to permit setting these data in the DØ loop over ISØIN

SØD(L)

Shock standoff distance divided by model radius (hemispheric L = 1, flat-faced L = 2)

SQTE

Square root of the shock density ratio, \( \sqrt{\varepsilon} \)

SQ2F(K)

Ratio of the stagnation-point heat flux for the current model geometry to that for an axisymmetric model (two-dimensional model K = 1, axisymmetric K = 2)

THI

\( T_2 \) value corresponding to enthalpy error ERRH, \( C_K \)

TITLE2

Hollerith data used in output

TILØ

\( T_2 \) value corresponding to enthalpy error ERRL, \( C_K \)

TSTAR

Reference temperature in flat-plate calculation, \( C_K \)

T1

Temperature ahead of the shock, \( C_K \)

T2

Temperature behind the shock, \( C_K \)

U1

Velocity ahead of the shock, cm/sec

VGP(L)

Velocity gradient parameter \( (R_g/u_1)(du'/dx) \)
(hemispherical model L = 1, flat-faced L = 2)

4.100 Subroutine NEWRAP

GJ(I)

Species concentrations, mole/g

I

DØ index; also J + ISC

IL

I + ISC
J  DØ index
K  DØ index
L  I + ISC
LL  ISSNR + 1
M2  ISC + 2
NØDEPS  Logical indicator; .TRUE. when the gas model contains no dependent species
RNAME  Routine name for calling DUMP routine
ZB  Correction factors $l+h_n^R$ to the mole fractions, eq. 1(260)
ZC  Correction factor $l+h_{C+1}^R$ to the pressure

4.101 Subroutine NEXTMP

ITS  Index of the next specified test section diameter
ØNE  1.0
XMØDL  Coordinate of the next model point in the geometric sequence of model points, cm
XMØD2  Coordinate of the model point at the next specified test section diameter, cm
XMØDEL  Coordinate of the next model point, cm

4.102 Subroutine NØNEQ

AB  Temporary storage location used in interchanging rows of $\beta_{ij}$ matrix during determination of matrix rank

-296-
ADCH  Absolute value of a $\delta \chi_1$ value

ALGOT  $\ln \ (10000)$

BLNEAR(M)  Array equivalence to the first 10 locations of common block /BLNE/

CA  BETA(M,I)/BETA(IX,I)

CAISV(I)  Storage for the correct values of CAI(I) to allow resetting CAI(I) when the artificial increase of reaction rate constants has been used to prevent premature switching from the perturbation technique to the numerical integration

CI(K)  Array specifying elemental composition of the gas mixture in the reservoir; CI(K) is the number of gram-atoms of the Kth element per gram of mixture

CJ  Number of gram-atoms of an element per gram of mixture at the end of a successful integration step

DCA(K)  \[ \frac{1}{(\gamma K')^2} \sum_{j=1}^{C} \delta c_j A_{jk} \] in eq. I(346c)

DCHMAX  Largest $|\delta \chi_1|$ for any reaction

DCHMIN  Smallest $|\delta \chi_1|$ for any reaction

DCI(K)  $\delta c_k$, correction in concentration of Kth chemical element required at the end of an integration step (eq. I(344), g-atom/g

DELIBIB(L)  Nondimensional boundary layer displacement thickness for the Lth profile, $\delta_{bl}/R_0$, at the start of an integration step
DELT: Nondimensional temperature interval in the perturbation solution, $|\Delta T|/T_0$

DELTSV: Input value for DELT1

DQEM: Maximum allowable change in the energy transfer to the electron gas in one integration step

DXM: Largest allowable step size for the first integration step after the switch from the perturbation solution

DXOLD: Step size $\Delta x$ in the preceding integration step (cm)

DXSAVE: Temporary storage for saving and restoring the integration step size when a reduced step is used to calculate the conditions at a model point (cm)

GJ(J): Concentration of the Jth species, $\gamma_j$, mole/g

GSQ(J): Square of GJ(J)

HDXSAV: One-half of DXSAVE, cm

I: $D\phi$ index

ICOUNT: Counter for number of times a step has been restarted with reduced step size

IERR: Error exit indicator (see subroutine listing)

IF(N): Array for storing and printing out IPFAIL values for successive restarts of a step (see glossary for common block /TNERK/)

II: I + ISC

IMAX: Index of the reaction with the largest $|\delta X_i|$
IRA  Counter for number of times the artificial increase in rate constants has been used to prevent premature switching from the perturbation technique to numerical integration.

ISMCP1  n - c + 1

ISTEPS  Counter for number of integration steps

ISUBD  Counter for number of times the temperature interval $\Delta T$ has been cut in half in an attempt to find a point in the perturbation solution satisfying the second inequality of eq. I(381)

ISUPR(I)  Array of indices of the reactions which are suppressed when the decrease of a minor species is found to control the integration step

IS5  Input value of ISW5B

ITAPE  Number of records written on the binary output tape since subroutine THR@AT was called

ITS  Index of the next model point defined by a specified test section diameter

IX  Rank of the $\beta_{ij}$ matrix

I2  Index used in calculating the rank of $\beta_{ij}$

J  Index used for various purposes

K  Index used for various purposes

IA  Index used in calculating the rank of $\beta_{ij}$

L  Dø index

M  Index used in calculating the rank of $\beta_{ij}$
N  Index used in calculating the rank of $\beta_{ij}$

NSUPR  Number of reactions suppressed when the decrease of a minor species is found to control the integration step

ØNE  1.0

ΩEØLD  Energy transfer to the electron gas at the preceding flow point, cal/cm$^3$ sec

RATIO  Ratio of the smallest and largest $|\Delta \gamma_i|$ values

RATIO2  Factor by which the forward rate constant is increased in the artificial increase of a reaction rate to prevent premature switching from the perturbation technique to the numerical integration

RNAME  Subroutine name for call to DUMP routine

SCD  Factor by which the step size $\Delta x$ is divided when an integration step fails and has to be restarted

SDGJ(J)  Change in the concentration $\gamma_j$ of the Jth species in an integration step

TENTH  10000.

TEP  Electron temperature ($^\circ$K)

TP  Gas temperature ($^\circ$K)

TSZ  Temporary storage for TPRINT

XMODEL  Position of next model point, cm

XMODEL1  Coordinate of the next model point in the geometric sequence of model points, cm
XMSET Logical indicator, set to .TRUE. when NEXTMP is first called to determine XMODEL

ZZZ Temporary storage

4.103 Subroutine NRMAX

DELT Decrement of nondimensional temperature

F1,F2,F3 Mass flux values in iteration to determine equilibrium sonic flow conditions

GJ(J) Species concentrations, mole/g

I $\text{D}\phi$ index over species

J $\text{D}\phi$ index over species

T1,T2,T3 Nondimensional temperature values in iteration to determine the equilibrium sonic flow conditions

Z $F_1 - F_2 + F_3 - F_2$

4.104 Subroutine OUT1

ANAM Hollerith word "ARAT"

ARAT Geometric area ratio

ASOLN(I,J) Hollerith array containing names for types of solution (see DATA statement)

ASTRSK Hollerith word filled with asterisks

CHDIMS(L) Transverse dimensions of a channel at the current flow point, inches

CP Molar heat capacity at constant pressure, divided by $R_0$
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DARAT</td>
<td>Derivative of the geometric area ratio</td>
</tr>
<tr>
<td>DEL</td>
<td>Electron density, electrons/cm³</td>
</tr>
<tr>
<td>DELSTP(L)</td>
<td>Displacement thickness on the Lth profile, cm</td>
</tr>
<tr>
<td>DGJ(I)</td>
<td>( \gamma_i' / dx ), derivatives of species concentrations, mole/g-cm</td>
</tr>
<tr>
<td>DLI:DX(I)</td>
<td>( \ln \gamma_i' / dx ) for the Ith species</td>
</tr>
<tr>
<td>DNAM</td>
<td>Hollerith work &quot;DIAM&quot;</td>
</tr>
<tr>
<td>EMF</td>
<td>Mole fraction of the electrons</td>
</tr>
<tr>
<td>FAREA</td>
<td>Effective cross-sectional area of the flow at the throat, cm²</td>
</tr>
<tr>
<td>FLX</td>
<td>Mass flux, lbm/ft²·sec</td>
</tr>
<tr>
<td>FVΩ2(I,J)</td>
<td>Array used for output of SN and XSN</td>
</tr>
<tr>
<td>GAMMA</td>
<td>Ratio of specific heats, ( \gamma = c_p / c_v )</td>
</tr>
<tr>
<td>GJ(I)</td>
<td>Species concentrations, mole/g</td>
</tr>
<tr>
<td>HOUT</td>
<td>Enthalpy, Btu/lb</td>
</tr>
<tr>
<td>HW</td>
<td>Enthalpy at the nozzle wall, Btu/lb</td>
</tr>
<tr>
<td>HO</td>
<td>Stagnation enthalpy, Btu/lb</td>
</tr>
<tr>
<td>I</td>
<td>DØ index</td>
</tr>
<tr>
<td>IBL</td>
<td>Index such that FVΩOUT(IBL) is the boundary layer displacement thickness on the nozzle wall or on the broad face of a channel</td>
</tr>
<tr>
<td>IFLAG</td>
<td>Indicator for entry points to subroutine</td>
</tr>
<tr>
<td>IFLAG = 1</td>
<td>Entry OUT1</td>
</tr>
<tr>
<td>IFLAG = 2</td>
<td>Entry OUT2(ISØIN)</td>
</tr>
</tbody>
</table>
II Index 7 + 10*I
INEQP1 INEQ + 1
ISOILN Indicator for type of solution
ISOILN = 1 Frozen
ISOILN = 2 Equilibrium
ISOILN = 3 Nonequilibrium
ISSSPL1 (ISS-1)/5+1, number of lines of output required for printing the species mole fractions
IS6 |ISW6B|
IX Counter of records written on tape 8 for a given flow solution
I1 Index of the first FVOUT(I) element to be printed on a line
I2 Index of the last FVOUT(I) element to be printed on a line
I6 Counter; number of flow points printed since last output of reaction rate data
J DØ index
K DØ index
L DØ index
LIM1 Index of first species mole fraction to be printed on a line
LIM2 Index of last species mole fraction to be printed on a line
PØUT Pressure, atm
PJ 3.1415927
RØUT  Density, lb/ft$^3$
SØUT  Entropy, Btu/lbm$^\circ$R
STEP(I)  Hollerith array (see DATA statement)
STFAC  $1/ [\rho u(h_0-h_\infty)]$, (Btu/ft$^2$-sec)$^{-1}$
TFLØW  Total mass flow, lb/sec
TNØAM(I)  Hollerith array of variable names (see DATA statement)
TNØUT(I)  Array used for output of quantities for electron non-equilibrium model
TØUT  Temperature, $^\circ$K
TYPSTN(I,J)  Hollerith array used to indicate whether non-equilibrium solution is based on perturbation technique (INEQ = 0) or numerical integration (INEQ = 1)
UØUT  Velocity, ft/sec
VARNAM(I)  Array of Hollerith names for output variables (see DATA statement)
VARN2(I)  Array of Hollerith names for N, XSN
WNAM  Hollerith word "WIDTH"
Y  Distance from nozzle axis to nozzle surface (or to first profile in the case of a channel)
Z  Distance from axis to second profile of a channel

4.105 Subroutine PERT

I  DØ index
J  DØ index
K  Dø index
Z  Temporary storage used in computing sums
Z1 Temporary storage used in computing sums

4.106 Function PIÔMEG

I  Index of entries in cross section table
PIÔMEG Cross section value returned to the calling routine

TE  Electron temperature, °K

4.107 Subroutine PRØP

J  Dø index over species
RNAME Routine name for call to DUMP

S4 \[ \frac{1}{R_0} \sum_{j=1}^{n} \gamma_j \sigma_j^0 \]

4.108 Subroutine PRTA

AS  Speed of sound, cm/sec

4.109 Subroutine PUTQIN

I  First index of species pair
IDBG Count to determine when debug output is produced
J  Second index of species pair
K  Option KQ to be used in step L of the computations
KK Type of cross section; first index of Q(KK,I,J)
L  Index of current step in cross section computations
IQ  Index of species pair in IQ, JQ array
IQ1 Index of first species pair in IQ, JQ array to be used in step L of computations
IQ2 Index of last species pair in IQ, JQ array to be used in step L of computations
M  Index of first parameter in V array to be used in step L of computations
\( \phi M(KK) \) Variable for temporary storage of cross sections computed in current step, \( \phi M(1) = \Omega(1,1) \), \( \phi M(2) = \Omega(2,2) \), \( \phi M(3) = B \cdot \Omega(1,1) \)
WROTE WROTE = .TRUE. indicates that debug output has been produced in the current call to PUTQIN
X  Electron partial pressure in atmospheres

4.110 Subroutine QCQL

C  Control variable. For \( C > 0 \), \( \ln (f \Lambda) \) is computed from eqs. I(100); for \( C \leq 0 \), \( \ln (f \Lambda) \) is set to 1
\( \phi M1 \) 0.8\( Q_C \) (see eq. I(100b))
X  Electron partial pressure in atmospheres
Y  20.91Y = (f \( \Lambda \))\(^2\) (see eqs. I(100c) and I(100d))

4.111 Subroutine QEX

A  Coefficient A in eq. I(104) for the exchange cross section
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B$</td>
<td>Coefficient $B$ in eq. I(104) for the exchange cross section</td>
</tr>
<tr>
<td>$\phi M(k)$</td>
<td>Contribution to cross sections from option: $\phi M(1) = \eta (3,1)$, $\phi M(2) = \eta (2,2)$, $\phi M(3) = B\eta (3,1)$</td>
</tr>
<tr>
<td>$V(M)$</td>
<td>Input parameters $VV$ for option</td>
</tr>
<tr>
<td>$Y$</td>
<td>Quantity $A - \frac{1}{4} B \ln (T/M)$</td>
</tr>
<tr>
<td>$ZM$</td>
<td>Molecular weight</td>
</tr>
</tbody>
</table>

**4.112 Subroutine QEXP**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Coefficient $A/k$ in $^0K$ for the exponential potential</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Quantity $\alpha$</td>
</tr>
<tr>
<td>$N1$</td>
<td>Position of first entry in tabulated collision integrals for the exponential potential</td>
</tr>
<tr>
<td>$\phi M(k)$</td>
<td>Computed cross section values (see subroutine QEX)</td>
</tr>
<tr>
<td>$\rho$</td>
<td>Characteristic length $\rho$ for the exponential potential in $\mathbb{R}$</td>
</tr>
<tr>
<td>$V(M)$</td>
<td>Input parameters $VV$ for option</td>
</tr>
</tbody>
</table>
4.113 Subroutine QINTRP

A Value of independent variable to be used in interpolation

B(J) Interpolated value of Jth dependent variable

I Interpolation is performed between entries I and I + 1 of the table

J Index of dependent variable

N Number of entries in table

N1 Location of first entry in table

N2 N2 + 1 is the location of the last entry in table

4.114 Subroutine QLJ

EPSIN Well depth parameter $\epsilon/k$ for the Lennard-Jones potential, in $\text{OK}$

N1 Position of first entry in tabulated cross sections for the Lennard-Jones potential

$\Phi_M(K)$ Computed cross section values (see subroutine QEX)
SIGMA  Collision diameter $\sigma$ for the Lennard–Jones potential, in Å

TSTAR  Nondimensional temperature, $kT/\epsilon$

V(M)  Input parameters VV for option

4.115 Subroutine QMTX

I  First index of species pair

J  Second index of species pair

K  Type of cross section, i.e., first index of $Q(K,I,J)$

IQ  Index of species pair in IQ, JQ array

IQ1  Index in IQ, JQ array of first species pair to be used in computation

IQ2  Index in IQ, JQ array of last species pair to be used in computation

SQT(K,I)  $\frac{1}{2} \sqrt{Q(K,I,I)}$

4.116 Subroutine QREP

ITL  Location in $\varnothing$MEGAL array where data are stored

$\varnothing$(K)  Computed cross section values (see subroutine $\varnothing$EX)

VV(M)  Input parameters VV for option

4.117 Subroutine QSAME

C  Constant by which previously computed cross sections are multiplied

I  First index of species pair for previously computed cross sections
J  Second index of species pair for previously computed cross sections

J\(\phi M(K)\)  Computed cross section values (see subroutine QEX)

V(M)  Input parameters VV for option

4.118 Subroutine QTAB

A  Constant multiplying factor for tabulated cross sections

N1  Location of first entry in tabulated cross section data

NL  Number of entries in tabulated cross section data

J\(\phi M(K)\)  Computed cross section values (see subroutine QEX)

V(M)  Input parameters VV for option

4.119 Subroutine QII

LQ1  Location in IQ, JQ list of the first species pair to which the ramp function is to be applied

LQ2  Location in IQ, JQ list of the last species pair to which the ramp function is to be applied

T0  Temperature in \(^{\circ}\)K beyond which cross sections are set to zero

T1  Temperature in \(^{\circ}\)K beyond which cross sections are unchanged

VI(M)  Input parameters VV for option
Input parameters for subroutine Q14

4.120 Subroutine Q12

I  
First index of first previously computed species pair

J  
Second index of first previously computed species pair

K  
First index of second previously computed species pair

L  
Second index of second previously computed species pair

M  
Index indicating cross section type

ΩM(M)  
Computed cross section values (see subroutine QEX)

Input parameters VV for option

4.121 Subroutine Q13

C  
Constant by which cross section is multiplied

I  
First index of species pair

J  
Second index of species pair

K1  
Type of cross section to be computed

K2  
Type of cross section to be used in computation

IQ  
Location of species pair in IQ, JQ list

IQ1  
Location in IQ, JQ list of first species pair to be used
1. **LQ2**
Location in IQ, JQ list of last species pair to be used

2. **VV(M)**
Input parameters VV for option

3. **4.122 Subroutine Q14**

4. **C2**
Constant applied to $\Omega^2 (2,2)$

5. **C3**
Constant applied to $B \Omega (1,1)$

6. **I**
First index of species pair

7. **J**
Second index of species pair

8. **LO**
Location of species pair in IQ, JQ list

9. **LQ1**
Location in IQ, JQ list of the first species pair to be used with the set of multiplying factors

10. **LQ2**
Location in IQ, JQ list of last species pair to be used with the set of multiplying factors

11. **VV(M)**
Input parameters VV for option

12. **4.123 Subroutine RADIUS**

13. **AG**
Geometric area ratio

14. **AGJ**
Factor proportional to $r_2^2$ in eq. I(171)

15. **ITYPE**
Indicator for type of nozzle geometry
   - ITYPE = 1 Two-dimensional nozzle
   - ITYPE = 2 Axisymmetric nozzle
   - ITYPE = 3 Channel

16. **L**
Index of profiles (1 for a nozzle, 1 or 2 for a channel)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>M</td>
<td>NPR(L)</td>
</tr>
<tr>
<td>NPR(L)</td>
<td>2 for ( L = 1 ); 1 for ( L = 2 )</td>
</tr>
<tr>
<td>R</td>
<td>Distance from axis to profile</td>
</tr>
<tr>
<td>RATIØ</td>
<td>( R/R_0 )</td>
</tr>
<tr>
<td>X</td>
<td>Axial coordinate in nozzle or channel, cm</td>
</tr>
<tr>
<td>YZ(L)</td>
<td>Distance from axis to ( L )th profile in a channel</td>
</tr>
</tbody>
</table>

4.124 **Subroutine READ**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Number of atoms of the ( K )th element per molecule of the ( I )th species, as obtained from the SPRP array</td>
</tr>
<tr>
<td>ASYM(IZ)</td>
<td>Hollerith chemical symbol for the element with atomic number ( I_2 )</td>
</tr>
<tr>
<td>CAPQ(I)</td>
<td>Number of gram atoms of the ( I )th chemical element per molecule of the cold gas</td>
</tr>
<tr>
<td>CP(I,J)</td>
<td>Array for addressing the channel data in common block /CHAN/</td>
</tr>
<tr>
<td>CSMW</td>
<td>Molecular weight of one of the cold species</td>
</tr>
<tr>
<td>DAL</td>
<td>Error in rounding ( A ) to an integer ( \text{LP} )</td>
</tr>
<tr>
<td>DUM(I)</td>
<td>Array for zeroing common block /TNCE/</td>
</tr>
<tr>
<td>ELNAME(K)</td>
<td>Hollerith name for the ( K )th element in one of the cold species</td>
</tr>
<tr>
<td>EPRP(I,J)</td>
<td>Array for addressing the element data in common block /ELEM/</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>EXPMP</td>
<td>Exponent used in computing the factor (FACMP) used to generate a geometric sequence of distances at which model calculations will be done</td>
</tr>
<tr>
<td>FNMP</td>
<td>Number of model points in geometric sequence, minus 1</td>
</tr>
<tr>
<td>GPRP(I,J)</td>
<td>Array for addressing the gas model data in common block /MIXT/</td>
</tr>
<tr>
<td>HWINØV</td>
<td>Estimate of stagnation enthalpy based on Winiwvich equation, Btu/lb</td>
</tr>
<tr>
<td>I</td>
<td>D∅ index</td>
</tr>
<tr>
<td>IES</td>
<td>Index of an element in a species, in the master list of elements</td>
</tr>
<tr>
<td>IGMV</td>
<td>Number of electronic levels in a species</td>
</tr>
<tr>
<td>IGS</td>
<td>Input value of IGAS</td>
</tr>
<tr>
<td>II</td>
<td>Index used in place of subscripted integer variables</td>
</tr>
<tr>
<td>IPRFL</td>
<td>Index used in place of subscripted integer variable, NPRØFL(J)</td>
</tr>
<tr>
<td>ISH</td>
<td>A value of ISHAPE(K,J)</td>
</tr>
<tr>
<td>ISKUR</td>
<td>Index of a third-body species in the master list of species</td>
</tr>
<tr>
<td>IUPDI</td>
<td>Initial value of IUPD</td>
</tr>
<tr>
<td>IZ</td>
<td>Atomic number of an element defined in the input using EEPRP</td>
</tr>
<tr>
<td>J</td>
<td>D∅ index</td>
</tr>
<tr>
<td>JDP1</td>
<td>JDIM + 1</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>JPS</td>
<td>Index of a species on the product side of a reaction, in the master list of species</td>
</tr>
<tr>
<td>JRS</td>
<td>Index of a species on the reactant side of a reaction, in the master list of species</td>
</tr>
<tr>
<td>K</td>
<td>DØ index</td>
</tr>
<tr>
<td>KK</td>
<td>Index of an element in a cold species, in the master list of elements</td>
</tr>
<tr>
<td>L</td>
<td>DØ index</td>
</tr>
<tr>
<td>LL</td>
<td>Index used in obtaining the PARAM array from the data stored in the ZPRP array</td>
</tr>
<tr>
<td>LP</td>
<td>Integer obtained by rounding AL</td>
</tr>
<tr>
<td>NATØM</td>
<td>Number of atoms of an element in a molecule of a species</td>
</tr>
<tr>
<td>NELS</td>
<td>Number of elements in a species</td>
</tr>
<tr>
<td>NK</td>
<td>Number of third body species for a reaction</td>
</tr>
<tr>
<td>NØZZLE</td>
<td>Index of the nozzle profile in the precoded geometry fits</td>
</tr>
<tr>
<td>NPSS</td>
<td>Number of product species for a reaction</td>
</tr>
<tr>
<td>NRSS</td>
<td>Number of reactant species for a reaction</td>
</tr>
<tr>
<td>NSECTD(J)</td>
<td>Number of downstream sections in the profile curvefit for the Jth profile of a channel or nozzle</td>
</tr>
<tr>
<td>NSECTJ</td>
<td>Index used in place of dimensioned integer variable, NSECT(J)</td>
</tr>
</tbody>
</table>
NSM1  NSECT(J) - 1
ØNE    1.0
PLUS   Hollerith + sign
RPRP(I,J) Array for addressing reaction data in common block /REAC/
SBJ(J)  b_j; eqs. I(51)
SETIGS Logical indicator; .TRUE. if automatic air model selection option is in effect
SHAPD  Array of Hollerith data for describing the shapes of nozzle profile sections
SHJAP(I) Formation enthalpy of the Jth species, cal/mole
SPNAME(I) Array of nominal species names for species defined in the input
SPRP(I,J) Array for addressing species data in common block /SPEC/
SY1(K)  Array of Hollerith left parentheses
SY2(K)  Array of Hollerith right parentheses
THEVP(I) Vibrational temperature for the Ith species, °K
TNEP(I,J) Array for addressing thermal nonequilibrium data in common block /TNE/
WORD(I,J) Hollerith data for output of the words "two-dimensional" and "axisymmetric"
WORD2(I) Hollerith data for output of the words "gap" and "diam"
ZPRP(I,J) Array for addressing the geometry data in common block /NØZZ/

4.125 Subroutine RESET

I Elapsed time since the beginning of the run in milliseconds

ICPU CPU time remaining before TIME, GØ, milliseconds

IN Entry point indicator: IN = 1, RESET; IN = 2, TIME

IZERØ CPU time remaining before TIME, GØ at call to RESET, milliseconds

4.126 Subroutine RSTMP

AM(I,J) Matrix of coefficients for system of equations in quadratic interpolation for the reservoir temperature

BM(I) Constant terms for system of equations in quadratic interpolation for the reservoir temperature

CØNST \( R_0 (m*/p_0)^2 \), in which \( R_0 \) is the universal gas constant, \( m* \) the known sonic mass flux, and \( p_0 \) the reservoir pressure, all expressed in cgs units

CTAPP Previous estimate of reservoir temperature, \( ^\circ K \)

CTSÅVE(I) For \( I = 1 \) to 3, successive estimates of the reservoir temperature, the value for \( I = 1 \) being the most recent, \( ^\circ K \)

CTSV Temporary storage for the reservoir temperature estimate used in the current iteration, \( ^\circ K \)

DCT The difference between the two most recent estimates of the reservoir temperature, \( ^\circ K \)
DCTAP: Absolute value of the difference between the current and improved estimates of the reservoir temperature, °K

DCTAP°: DCTAP in the previous iteration step, °K

DMC: Intermediate quantity in quadratic interpolation for the reservoir temperature, °K

D1,D2: Intermediate quantities in linear interpolation for the reservoir temperature, °K

F(I): For I = 1 to 3, successive estimates of the quantity W0·(SM)^2, the value for I = 1 being the most recent; W0 is the molecular weight in the reservoir and SM the nondimensional sonic mass flux

H: Calculated value of reservoir enthalpy, cal/g

HERR: Allowable error in calculated stagnation enthalpy cal/g

H°: Calculated stagnation enthalpy in previous iteration step, cal/g

I: D° index

J: Index for shifting data in CTSAVE and F arrays

KS: Indicator for singularity of AM matrix

N: Iteration counter for main iteration

N2: Iteration counter for temperature calculation in enthalpy option

N3: Iteration counter for pressure calculation in enthalpy option
<table>
<thead>
<tr>
<th>PCGS</th>
<th>Reservoir pressure, dyne/cm²</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERR</td>
<td>Allowable relative error in reservoir pressure</td>
</tr>
<tr>
<td>PIAST</td>
<td>Previous estimate of reservoir pressure, atm</td>
</tr>
<tr>
<td>RGAS</td>
<td>Universal gas constant in cgs units</td>
</tr>
<tr>
<td>RNAME</td>
<td>Hollerith name for calling DUMP routine</td>
</tr>
<tr>
<td>SMERR</td>
<td>Allowable error in sonic mass flux, g/cm² sec</td>
</tr>
<tr>
<td>SMP</td>
<td>Calculated sonic mass flux, g/cm² sec</td>
</tr>
<tr>
<td>TERR</td>
<td>Allowable relative error in reservoir temperature</td>
</tr>
<tr>
<td>TLAST</td>
<td>Previous estimate of reservoir temperature, °K</td>
</tr>
</tbody>
</table>

4.127 **Subroutine RNKT**

**D*J(J)**  
\(d \gamma_j/dx\), derivatives of species concentrations, mole/g-cm

**DQMRK**  
Twice DQMAX; criterion value for allowable changes in energy transfer to the electrons, QDPE, cal/cm³ sec

**FK(J,K)**  
Array equivalenced to F1(J), F2(J), F3(J), F4(J)

**F1(J)**  
\(f_1\), derivative of the Jth dependent variable at the start of the integration step

**F2(J)**  
\(f_2\), derivative of the Jth dependent variable at \(x_2,y_2\) (eq. I(394a))

**F3(J)**  
\(f_3\), derivative of the Jth dependent variable at \(x_3,y_3\) (eq. I(394b))
F4(J) \( f_j \), derivative of the \( J \)th dependent variable at \( x_4 \), \( y_4 \) where \( y_4 \) is given by I(394c) or I(405)

GJK(J,K) Array equivalence to GJ1(J), GJ2(J), GJ3(J), GJ4(J)

GJ1(J) \( y_1 \), the \( J \)th dependent variable at the start of the integration step

GJ2(J) \( y_2 \), the \( J \)th dependent variable at \( x_2 \) (eq. I(394a))

GJ3(J) \( y_3 \), the \( J \)th dependent variable at \( x_3 \) (eq. I(394b))

GJ4(J) \( y_4 \), the \( J \)th dependent variable at \( x_4 \) (eq. I(394c) or I(405))

IFØR Indicator for type of formula used to compute GJ4(J); IFØR=1 for I(394c), IFØR=2 for eq. I(405)

ISTMNT Indicator for locations in subroutine at which diagnostic dumps are written (see listing)

\( J \) Index of dependent variables:
\[
\begin{align*}
J = 1 & \text{ to } n & \gamma_j \\
J = n + 1 & \text{ to } J & T_j \\
J = n + 2 & & T_e \\
J = n + 3 & & T_0
\end{align*}
\]

K DØ index

LIM Number of dependent integration variables

P(J) Parameter (-P \( Ax \)) for the \( J \)th dependent variable, eq. I(402)

QDPEB Electron energy transfer (QDPE) at the start of the integration step, cal/cm\(^3\) sec

SDGJ(J) Change in \( \gamma_j \) over an integration step, mole/g
<table>
<thead>
<tr>
<th>SDQ(J)</th>
<th>Change in Jth dependent variable over the integration step</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Temporary storage in computation of I(398a) for small PΔx</td>
</tr>
<tr>
<td>S2</td>
<td>Temporary storage in computation of I(398b) for small PΔx</td>
</tr>
<tr>
<td>S3</td>
<td>Temporary storage in computation of I(398c) for small PΔx</td>
</tr>
<tr>
<td>TE</td>
<td>Logical indicator, .TRUE. for chemical non-equilibrium models, .FALSE. for electronic nonequilibrium models</td>
</tr>
<tr>
<td>T1</td>
<td>Temporary storage in computation of I(398a) for small PΔx</td>
</tr>
<tr>
<td>T2</td>
<td>Temporary storage in computation of I(398b) for small PΔx</td>
</tr>
<tr>
<td>T3</td>
<td>Temporary storage in computation of I(398c) for small PΔx</td>
</tr>
<tr>
<td>XX</td>
<td>( \log_{10}(</td>
</tr>
<tr>
<td>XL</td>
<td>( F_1 ) (eq. I(398a))</td>
</tr>
<tr>
<td>X2</td>
<td>( F_2 ) (eq. I(398b))</td>
</tr>
<tr>
<td>X3</td>
<td>( F_3 ) (eq. I(398c))</td>
</tr>
<tr>
<td>X4</td>
<td>( f_2-f_1 )</td>
</tr>
<tr>
<td>Y(K)</td>
<td>Array equivalenced to Y1, Y2, Y3, Y4</td>
</tr>
<tr>
<td>YY</td>
<td>( \log_{10}(</td>
</tr>
<tr>
<td>Y1</td>
<td>( f_1 + P_{Y1} ) in I(401)</td>
</tr>
<tr>
<td>Y2</td>
<td>$f_2 + Py_2$ in I(401)</td>
</tr>
<tr>
<td>------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Y3</td>
<td>$f_3 + Py_3$ in I(401)</td>
</tr>
<tr>
<td>Y4</td>
<td>$f_4 + Py_4$ in I(401)</td>
</tr>
</tbody>
</table>

4.128 **Subroutine SHOCK**

**ALP**
- The quantity $\sin^2 \sigma - B$, where $\sigma$ is the shock angle.

**B**
- The quantity $- [\frac{M^2+2}{M^2} + \frac{\gamma \sin^2 \alpha}{M^2}]$, where $M$ is the free-stream Mach number, $\gamma$ the specific heat ratio, and $\alpha$ the body angle.

**BETA**
- The quantity $\text{ALP} \cdot \sin^2 \sigma + C$

**BMF**
- Free-stream Mach number, $M$

**BM2**
- $M^2$, square of the Mach number

**BM4**
- $M^4$

**C**
- The quantity $\frac{2M^2+1}{M^4} + \left[ \frac{1}{4} (\gamma+1)^2 + \frac{\gamma-1}{M^2} \right] \sin^2 \alpha$

**CD**
- Cosine of the body angle, $\cos \alpha$

**C1**
- $\gamma + 1$

**C2**
- $\gamma - 1$

**C3**
- $2B$

**D**
- $-\cos^2 \alpha \div M^4$

**DEL**
- The body angle $\alpha$; inclination of the wedge surface to the direction of free-stream motion (degrees)

**DELR**
- $\alpha$ in radians

-322-
\[ \sqrt{(ALP)^2 - 4 \cdot BETA} \]

**FL(I)** Array used for communicating results to the calling routine

- FL(1) shock angle
- FL(2) Mach number behind shock
- FL(3) Density ratio, \( \rho_2 / \rho_1 \)
- FL(4) Static pressure ratio, \( p_2 / p_1 \)
- FL(5) Static temperature ratio, \( T_2 / T_1 \)
- FL(6) Total pressure ratio, \( P_{t2} / P_{t1} \)
- FL(7) Ratio, \( T_{stag} / T_1 \)

**FPU** Derivative \( dF/du \) of the function \( F(u) \)

**FU** \( F(u) = u^3 + Bu^2 + Cu + D \)

**GAM** Specific heat ratio \( \gamma \)

**IERR** Argument whose value is set to 2 when Newton-Raphson iteration to find the shock angle does not converge

**KIT** Iteration counter

**RT** \( \sin^2 \sigma \), where \( \sigma \) is the shock angle

**SD** \( \sin \alpha \), where \( \alpha \) is the body angle

**UG** Previous trial value of \( u \) in solution of cubic equation \( F(u) = 0 \)

**UN** Current trial value of \( u \) in solution of cubic equation \( F(u) = 0 \)

**U1** First solution for \( \sin^2 \sigma \)

**U2** Second solution for \( \sin^2 \sigma \)

**V1** \( M^2 \sin^2 \sigma \)
### 4.129 Subroutine SIMQ

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(I)</td>
<td>Matrix of coefficients for the system of equations to be solved, stored columnwise with no gaps</td>
</tr>
<tr>
<td>B(I)</td>
<td>Constant terms for the system of equations to be solved; these data are replaced by the solution values</td>
</tr>
<tr>
<td>BIGA</td>
<td>Pivotal (largest) element in a column</td>
</tr>
<tr>
<td>I</td>
<td>Row index</td>
</tr>
<tr>
<td>IA</td>
<td>Index of matrix A in back solution</td>
</tr>
<tr>
<td>IB</td>
<td>Index of recomputed array B in back solution, containing solution values</td>
</tr>
<tr>
<td>IC</td>
<td>Index of array B in back solution</td>
</tr>
<tr>
<td>IJ</td>
<td>Index in search for pivotal element in column</td>
</tr>
<tr>
<td>IMAX</td>
<td>Index of pivotal element in column</td>
</tr>
<tr>
<td>IQS</td>
<td>Index in elimination of variables</td>
</tr>
<tr>
<td>IT</td>
<td>Index used for several purposes</td>
</tr>
<tr>
<td>IX</td>
<td>DØ index in elimination of variables</td>
</tr>
<tr>
<td>IXJ</td>
<td>Index in elimination of variables</td>
</tr>
<tr>
<td>IXJX</td>
<td>Index in elimination of variables</td>
</tr>
<tr>
<td>I1,I2</td>
<td>Indices in interchange of rows</td>
</tr>
<tr>
<td>J</td>
<td>Column index</td>
</tr>
<tr>
<td>JJ</td>
<td>Index of the first element in a column of the reduced matrix</td>
</tr>
</tbody>
</table>
JJX  Index in elimination of variables
JX   D∅ index in elimination of variables
JY   $J + 1$
K    D∅ index in interchange of rows
KS   Error indicator; set to 1 if matrix of coefficients is singular
N    Number of equations to be solved
NY   $N - 1$
SAVE Temporary storage location used in interchanging rows and eliminating variables
T∅L  Criterion for value of pivotal element in test for singular matrix

4.130  Subroutine STUNTS
ASTRSK Hollerith word filled with asterisks
DELT(I) Array of preset temperature intervals for table of species thermal properties, °K
HOO  Heat of formation of a species in kcal/mole
I    D∅ index; also value of IQ(IQ); also value of ISEQ(L)
ICARD Counter of punched cards containing cross section data
II   Index of preset temperature intervals for thermal property table
IIQJ Index of the first species of a pair of species whose cross section is computed in the Ith step of the transport property calculation
IT  Counter for lines in the thermal property table with the same value of the temperature increment

IZ  Number of lines in thermal property table

J  DØ index; also JQ(LQ)

JJQJ  Index of the second species of a pair of species whose cross section is computed in the Ith step of the transport property calculation

K  DØ index

KK  KQ(L)

L  Index of steps in the cross section calculation

LIM  1 for species with no thermo fit; 2 for species with a thermo fit

LIM1  1 for species with physical model data; 2 for species without such data

IQ  DØ index

LQ1  Index of the first species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied

LQ2  Index of the last species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied

M  DØ index

MV1  Index of the first parameter (in the V array) used in the Lth step of the edited cross section computation

MV2  Index of the last parameter (in the V array) used in the Lth step of the edited cross section computation
NDT(I)  Number of lines in the thermal property table for which the Ith preset temperature increment is used

NNQI  Number of species pairs to which the cross section calculated in the Ith step of the calculation is applied

TPRΩP(I,K,M)  Computed thermal properties for the Ith line of the table; M = 1 contains the results of the physical model, M = 2 the thermo-fit results (if any)

K = 1  Chemical potential
K = 2  Enthalpy
K = 3  Specific heat
K = 4  Entropy

TT(I)  Temperature value for the Ith line of the thermal property table, °K

TV  Temperature value, °K

4.131 Subroutine THERM

CCPTF  Nondimensional specific heat based on the thermo fit

CLGT  Natural logarithm of the nondimensional temperature, ln(T/T₀)

El  Logical variable, .TRUE. if the routine was entered through the entry point THERM1, otherwise .FALSE.

J  Species index

K  Index of vibrational states in a linear triatomic species

M  Index of the electronic states in a species
MIX Logical variable, set to .TRUE. if the species thermal properties are to be calculated by "mixing" results from the physical model and the thermo fit

N Number of electronic states in a species

SHTF Nondimensional enthalpy based on the thermo fit

STF Nondimensional entropy based on the thermo fit

SZT Temporary storage for a value required in the entropy calculation for a linear triatomic species

S1 See eq. I(56a)

S2 See eq. I(56c)

S3 See eq. I(56d)

WTHØ Weight of the physical model results in the "mixed" calculation of species properties

WTTF Weight of the thermo fit results in the "mixed" calculation of species properties

WW $\left( S_2 \right)^2$

XMTF Chemical potential based on the thermo fit results

XX Dummy variable used in calculations of electronic state contributions to species properties

ZT Difference between the temperature and the switchover temperature CTMXXI from the physical model to the thermo fit, °K

ZL-Z9 Storage for intermediate values in calculations of the vibrational contributions to species properties
4.132 Subroutine $THR\Theta AT$

AG  Geometric area ratio
I  DØ index
ONE  1.0
XX  Dummy argument for call to AGSØLN

4.133 Subroutine $TRANSP$

C(I,J)  Molecular weight dependent factor used in transport calculations (see eqs. 171)
CMSAVE  Mean molecular weight $W$ for the mixture
CPTØT  Nondimensional frozen specific heat at constant pressure for the mixture, $W c_{pf}/R_0$
CSAVE(I)  Array for saving previously computed values of the nondimensional specific heats $CCPJ(I)$
CTSAVE  Variable for saving previous value of the temperature ratio CT
I  Species index
J  Species index
J1  $J - 1$
P  Gas pressure in atmospheres
S  $\sqrt{TTAB}$
SSIG  Factor used in computation of electrical conductivity, $(e/k)^2/\sqrt{T}$
TTAB  Temperature in °K for which transport calculations are to be made
ZK(L) See eq. (173)
ZKINT See eq. (173)
ZM1(I) Factor proportional to the molecular weight of species I (see eq. (171))

4.134 Subroutine WEDGE

A \( \frac{(\gamma + 1)}{2} \), where \( \gamma \) = specific heat ratio
AE Free-molecule accommodation coefficient for heat transfer
ALMIN Minimum angle of attack for calculation of free-molecule heat flux (0.01 radian)
ALPHA Angle of attack in radians
AM Free-stream Mach number \( M \)
AMD Double-precision Mach number for call to SHOCK routine
AM2 \( M^2 \)
AND Input angle of attack in degrees or ANMIN, whichever is larger
ANMIN Minimum angle of attack for oblique shock calculation (10^{-4} degree)
APR(IA) Static pressure ratio, \( p_2/p_1 \), based on the oblique shock calculation for the IIAth angle
AST(J) Hollerith array for marking heat fluxes which are at the free-molecule limit with an asterisk
ASTQ(IT) Hollerith blank or asterisk for ITth column of output table
A2 \( A^2 \)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A4</td>
<td></td>
</tr>
<tr>
<td>A5</td>
<td></td>
</tr>
<tr>
<td>A7</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>[ \left( \frac{\mu(T)}{\mu(T_0)} \right) \cdot \frac{T_1}{T^*} ]; see Sec. 8.2.2, Vol. I</td>
</tr>
<tr>
<td>CAPGAM</td>
<td>Angle-of-attack parameter ( \Gamma ) (eq. I(481c) or I(501))</td>
</tr>
<tr>
<td>CH</td>
<td>Heat transfer coefficients, ( C_H )</td>
</tr>
<tr>
<td>CHD</td>
<td>Coefficient 0.332 ( A \in \tau N^2 t ) for calculating ( C_H ) from ( S^* )</td>
</tr>
<tr>
<td>CHDFAC</td>
<td>Coefficient 0.332 ( A )</td>
</tr>
<tr>
<td>CPGM</td>
<td>Single-precision value equal to ( \Gamma )</td>
</tr>
<tr>
<td>DELSTW(IT)</td>
<td>Boundary layer displacement thickness at the ( IT^{th} ) point along the wedge</td>
</tr>
<tr>
<td>DFAC</td>
<td>Coefficient ( \gamma - \kappa^3/8 \ A^5 \varepsilon )</td>
</tr>
<tr>
<td>DFAK</td>
<td>Quantity ( DFAC \cdot \frac{t}{(\tau N)^4} )</td>
</tr>
<tr>
<td>EN</td>
<td>Parameter ( N = M \sqrt{C/Re_t} )</td>
</tr>
<tr>
<td>EPS</td>
<td>Quantity ( \varepsilon = (\gamma - 1)/(\gamma + 1) )</td>
</tr>
<tr>
<td>EPS2</td>
<td>( \varepsilon^2 )</td>
</tr>
<tr>
<td>ERFV</td>
<td>Quantity ( 1 + \text{erf}(S \sin \alpha) ) used in calculating free-molecule limit to heat flux</td>
</tr>
<tr>
<td>EXPFV</td>
<td>Quantity ( e^{-(S \sin \alpha)^2} ) in free-molecule heat flux calculations</td>
</tr>
</tbody>
</table>
**FINAL**

Logical variable, set to .TRUE. when calculations have been done for all of the specified distances from the leading edge for given angle of attack and nose radius.

**FL(K)**

Array used by subroutine SHOCK to communicate its results to WEDGE. See glossary of symbols in SHOCK for definitions of the elements of FL.

**GAMMA**

Specific-heat ratio, $\gamma$.

**GAMD**

Double-precision value of GAMMA, for call to SHOCK routine.

**GFAC**

Factor used in calculating $\Gamma$; equal to $k/2A^2e M$ for the modified Cheng-Kemp theory, and to $\gamma k/2 A^2 e$ for the unmodified theory.

**GFAK**

$\text{GFAC}/(\tau N)^2$.

**GFC**

Coefficient $k/2A^2e$ in GFAC.

**GMW**

Mean molecular weight of gas, g/mole.

**G2**

$\gamma^2$.

**G3**

$\gamma^3$.

**HRATI\O**

Ratio $h_w/h_0$ of enthalpy at the wall to the free-stream stagnation enthalpy.

**HW**

Gas enthalpy at the wall temperature, cal/g.

**HO**

Free-stream stagnation enthalpy, cal/g.

**IA**

Index of angles of attack.

**IAMAX**

Index of angle of attack at which classical oblique shock solution gives a detached shock.
IERR  Error indicator for classical oblique shock solution, set in SHOCK routine:
         0  Detached shock
         1  Attached shock – normal case
         2  Convergence failure

IFM   Control variable; reset from 0 to 1 if any heat flux value is reduced to the free-molecule limit

IR    Index of leading edge radii

IS9   Absolute value of ISW9B

IT    Index of columns in the output table

ITAB  Index of blocks in the output table

ITH   Index equal to 1 for the modified Cheng-Kemp theory, equal to 2 for the unmodified theory

ITHS  Value of ITH before resetting because of IERR = 0

J     Implied DΘ index in WRITE statements

NW1   Index of points in the uniform sequence of distances from the leading edge

NW2   Index of specified distances from the leading edge

ØMEGA Parameter Ω, eq. I(502)

PFAC  Quantity 4 A² e² M²/γ k²

PFAK  Coefficient PFAC · (τ N)⁴ in formula for pressure on wedge

PI    Value 3.1415927

PINF  Free-stream pressure, atm
| PRATIØ | Ratio of pressure on wedge to free-stream pressure, $p_w/p_1$ |
| PW.(IT) | Pressure on the wedge at the ITth distance from the leading edge, atm |
| QFM(IA) | Free-molecule limit to the heat flux for the IAth angle of attack, Btu/ft²-sec |
| QFMC1 | Coefficient $0.08 \frac{S p_1 a_e}{\sqrt{RT_0/2 \pi W}}$ in calculation of free-molecule heat flux, eq. I(512) |
| QFMC2 | Coefficient $S^2 + \left[ \gamma - \frac{1}{2} (\gamma+1) \frac{T_w}{T_0} \right] / (\gamma-1)$ in free-molecule heat flux calculation, eq. I(512) |
| QWW(IT) | Heat flux to the wedge at the ITth distance from the leading edge, Btu/ft²-sec |
| REPF | Reynolds number per foot |
| RGC | Gas constant $(8.314 \times 10^7$ erg/mole $^o$K) |
| RHØINF | Free-stream density, $\rho_1$ |
| RNAME | Hollerith routine name for call to subroutine DUMP |
| RU2 | Dynamic pressure $\rho_1 u_1^2$ |
| S | Free-molecule flow parameter $M \sqrt{\gamma/2}$ |
| SQT | Quantity $\sqrt{z z' + \frac{\Omega^2}{S}}$ in modified Cheng-Kemp calculation; $\sqrt{z z'}$ in unmodified calculation |
| SQTP1 | $\sqrt{\pi}$ |
| SSA | $S \sin \alpha$ |
| SSASQ | $(S \sin \alpha)^2$ |
TAU
Parameter $\tau = 0.664 + 1.72 \frac{h_w}{h_0}$

TAUN $\tau N$

TAUN2 $(\tau N)^2$

TAUN3 $(\tau N)^3$

TAUN4 $(\tau N)^4$

TAUN6 $(\tau N)^6$

TINF Free-stream temperature, $T_1$

TN Diameter of leading edge, inch

TRATIO Ratio of wedge surface temperature to free-stream temperature

TREF Reference temperature, $(1 + 3 \frac{h_w}{h_0}) T_0 / 6$, where $T_0$ is the free-stream stagnation temperature behind a frozen normal shock

UNIF Free-stream velocity, $u_1$

UN(J) Hollerith array for labelling the output as being based on the modified or unmodified Cheng-Kemp theory

VMUINF Free-stream viscosity, lb/ft-sec

WK2 Square of leading-edge drag coefficient, $k^2$

WK3 $k^3$

WK4 $k^4$

WK51 Next distance from the leading edge in the specified uniform sequence, inch
<table>
<thead>
<tr>
<th>Term</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WXV2</td>
<td>Next separately specified distance from the leading edge, inch</td>
</tr>
<tr>
<td>XC</td>
<td>Coordinate x parallel to the free-stream flow for a point on the surface of the wedge at a distance XW(IT) from the leading edge; XC = XW(IT) \cdot \cos \alpha, inch</td>
</tr>
<tr>
<td>XM</td>
<td>Distance from the leading edge up to which merging effects are important (inch); see eq. I(510a)</td>
</tr>
<tr>
<td>XS</td>
<td>Distance from the leading edge beyond which the strong-interaction approximation breaks down (inch); see eq. I(510b)</td>
</tr>
<tr>
<td>XW(IT)</td>
<td>The ITth specified distance from the leading edge, inch</td>
</tr>
<tr>
<td>YS(IT)</td>
<td>Shock ordinate at the ITth distance from the leading edge, inch</td>
</tr>
<tr>
<td>YSFAC</td>
<td>Coefficient (\gamma^2 \kappa^3 / 8 \ A^4 \epsilon) in calculation of shock ordinate</td>
</tr>
<tr>
<td>YSFAK</td>
<td>Coefficient YSFAC \cdot t / (\gamma N)^4 in calculation of shock ordinate</td>
</tr>
<tr>
<td>Z</td>
<td>Nondimensional shock ordinate z; eq. I(481a)</td>
</tr>
<tr>
<td>ZERØ</td>
<td>0.0</td>
</tr>
<tr>
<td>ZETA</td>
<td>Nondimensional coordinate (\zeta) parallel to free-stream flow; see eq. I(481b)</td>
</tr>
<tr>
<td>ZFAC</td>
<td>Coefficient (16 \ A^7 \epsilon / \gamma^3 k^4)</td>
</tr>
<tr>
<td>ZFAK</td>
<td>Coefficient ZFAC \cdot (\gamma N)^6 / t</td>
</tr>
<tr>
<td>ZTA(IT)</td>
<td>(\zeta) at the ITth distance from the leading edge</td>
</tr>
<tr>
<td>ZZP</td>
<td>zz'</td>
</tr>
<tr>
<td>ZZPP</td>
<td>(zz')'</td>
</tr>
</tbody>
</table>

-336-
zzps z as a single-precision value

4.135 subroutine wesqln

b0 coefficient \( \frac{3}{\sqrt{6}} \) in initial approximation to \( \lambda \)

bl coefficient \( \sqrt{3} \) in initial approximation to \( \lambda \)

called logical variable used to bypass resetting of the "c" coefficients after the first call to wesqln

capgam angle of attack parameter, \( \Gamma \)

cgsq \( \Gamma^2 \)

c1 1. the "c" symbols are coefficients in the cheng-kemp analytical solution of the cheng equation for \( \Gamma = 0 \)

c2 2.

c3 \( 1/3 \)

c4 4.

c9 9.

c32 1.5

c103 \( 10/3 \)

c229 \( 22/9 \)

c463 \( 46/3 \)

c769 \( 76/9 \)

dldzt \( d \lambda / d \zeta \) for newton-raphson iteration
N  Iteration counter
P1  \( \zeta \ 1/3 \)
P2  \( \zeta \ 1/2 \)
P12 \( \zeta \ 2/3 \)
Q1  \( 1 + \sqrt{\lambda} \)
Q2  \( (1 + \sqrt{\lambda})^2 \)
Q3  \( (1 + \sqrt{\lambda})^3 \)
Q4  \( (1 + \sqrt{\lambda})^4 \)
Q5  \( \ln (1 + \sqrt{\lambda}) \)
SQTL  \( \sqrt{\lambda} \)
XL  \( \lambda \)
Z  Nondimensional shock ordinate, \( z \)
ZETA Nondimensional coordinate \( \zeta \) parallel to the direction of free-stream flow
ZT  \( \zeta \) calculated using trial value of \( \lambda \)
ZZP  \( zz' \)
ZZPP  \( (zz')'' \)

4.136 **Subroutine XSECT**

Because a number of symbols are used for different purposes in different sections of subroutine XSECT, the symbols used in each section of the subroutine are defined separately below.
**Statements to Statement 40**

I(J) Revised index for species J in master species list (common variable)

J Index of species in master species list

JJ Revised index for species not included in transport calculations

K Count of species

L Revised index for species included in transport calculations

**Statements 40 to 110**

J Species index

K Index of steps in the cross section calculation in the order in which they are carried out

KIJ Index in packed lists of species pairs to which edited cross section data are applied

KM Index of parameters for a single step of the cross section calculation

KV Index in packed list of parameter values for all steps of the cross section calculation

L Species index; step of cross section computations

LL First index of species pair

LO NQ(L)

M Index of the Kth step in the cross section calculation, equal to ISEQ(K)
MMV  Total number of parameters in V array

NK  Index of the option to be used in the Kth step; also, number of parameters used in step

Statements 110 through 190

I(J)  Revised index for species J in master list (common variable)

II  First index of unused species pair

III  II-1

J  Second index of species pair referenced in V array; species index in master species list

JJ  Index of species added to cross section computation

K  Index in master species list of species added to cross section computations

L  Index of step in cross section computations

LL  First index of species pair referenced in V array

IQ  Location of species pair in IQ, JQ array

IQ1  Location in IQ, JQ array of first species pair for step

IQ2  Location in IQ, JQ array of last species pair for step

MV  Dummy variable (not used)
Statement 190 to Card XSE 213

**IV**  First index of species pair computed with option $KQ = 10$

**J**    Second index of species pair to be moved

**JV**   Second index of species pair computed with option $KQ = 10$

**K**    Count of array elements; dummy species index

**L**    Step in cross section computations; dummy species index

**LL**   First index of species pair to be moved

**IQ**   Location of species pair IV, JV in IQ, JQ array

**IQ1**  Location in IQ, JQ array of last species pair for step $L-1$ of computations

**IQ2**  Location in IQ, JQ array of last species pair for step $L$ of computations

**LV**   Index of array elements to be moved

**MM**   Location in V array of last parameter for step $L-1$ of computations

**MMV**  Total number of parameters in V array

**MV**   Index for parameters in V array

**NSV**  Total number of steps in edited cross section computations

-341-
Card XSE 213 to Statement 300

II First index of unused storage location for species pair
IV First index of a species pair
J Second index of species pair to be moved
JJ Second index of unused storage location for species pair
JV Second index of a species pair
L Step of cross section computations
LL First index of species pair to be moved
LQ Index of species pair in IQ, JQ array
LQ2 Total number of species pairs in IQ, JQ array
MV Dummy argument, value not required

Statement 380 to end of subroutine

I(K) Species for which cross section data are to be supplied (common variable)
IE For IE = 1, cross section data are not supplied for electrons
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>II</td>
<td>Absolute value of IZ for a species</td>
</tr>
<tr>
<td>IZ(L)</td>
<td>Charge of species L in units of electron charge (Note that IZ is negative for positively charged species)</td>
</tr>
<tr>
<td>J</td>
<td>Index indicating charge for species for which cross sections are to be supplied (see Table III)</td>
</tr>
<tr>
<td>JJ</td>
<td>Index indicating whether cross sections are to be supplied for electrons (see Table III)</td>
</tr>
<tr>
<td>KK</td>
<td>Index indicating charge of species for which cross sections are to be supplied (see Table III)</td>
</tr>
<tr>
<td>L</td>
<td>Step in cross section computations; species index</td>
</tr>
<tr>
<td>LL</td>
<td>Computational step at which additional cross section data are supplied for neutral-neutral interactions</td>
</tr>
<tr>
<td>NEWMV</td>
<td>Total number of parameters in V array</td>
</tr>
<tr>
<td>NN</td>
<td>Number of species for which cross section data are to be supplied</td>
</tr>
</tbody>
</table>
Parameters $V(M)$ for effective Coulomb cross sections between two species with the electronic charges $KK$ and $J$, from eqs. I(100a). The parameters for $JJ = 1$ are for ion-ion collisions, and for $JJ = 2$ for ion-electron collisions.
5. REFERENCES TO COMMON VARIABLES

This section presents a computer-generated analysis of references to common variables and common-equivalent variables in the NATA code. The variables are listed in alphanumeric order. For each variable, the name of the common block containing it is given. The block name / refers to unlabelled common. An asterisk following the block name indicates that the variable does not appear in common, but is equivalenced to a common variable in the designated block in one or more routines. The routines in which each variable is referenced are indicated by numbers, based on the list of routine names given on the first page. Routine number 2 is the block data routine containing species, reaction, geometry, and other data; number 3 is the transport cross section block data routine. The routines EXP and RESET are missing from the list because the analysis was carried out for the UNIVAC version of NATA, which does not contain these routines. Neither EXP nor RESET contains any common blocks.

Lines containing a blank variable name are continuations of the preceding lines, in cases in which the number of referencing routines is too large to be printed on a single line.

Appearance of a variable in a common or equivalence statement is not considered to constitute a reference. A routine is listed as referencing a given variable if the variable appears in an arithmetic or DATA statement, a READ or WRITE statement or a NAMELIST statement, or is used as an index or a subroutine or function argument.

In the case of a common-equivalent variable, the program used to prepare the list checks for the presence of the associated common block in the routine, but does not verify the presence of the equivalence statement in the routine. For this reason, some of the listed references to common-equivalent variables are spurious. For example, in subroutine BLAYER, the variable X is equivalenced to the
variable CX in unlabelled common. Subroutines GEØMAR, ØUT1, and TRANSP are also listed as referencing X. All three of these references are spurious because X is not equivalenced to a common variable in these three routines.

In some cases, different names are used for the same common variable in different routines. All such names are included in the list below, each with its own list of referencing routines.
<table>
<thead>
<tr>
<th>Index</th>
<th>Routine</th>
<th>Index</th>
<th>Routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MAIN</td>
<td>36</td>
<td>NRMAX</td>
</tr>
<tr>
<td>2</td>
<td>BLOCKD</td>
<td>37</td>
<td>OUT</td>
</tr>
<tr>
<td>3</td>
<td>BLOCKD</td>
<td>38</td>
<td>OUTI</td>
</tr>
<tr>
<td>4</td>
<td>AESOLN</td>
<td>39</td>
<td>PERT</td>
</tr>
<tr>
<td>5</td>
<td>AESOLN</td>
<td>40</td>
<td>PIOMEG</td>
</tr>
<tr>
<td>6</td>
<td>AXFIT</td>
<td>41</td>
<td>PROP</td>
</tr>
<tr>
<td>7</td>
<td>AXSFCT</td>
<td>42</td>
<td>PRTA</td>
</tr>
<tr>
<td>8</td>
<td>PLAYER</td>
<td>43</td>
<td>PTOIN</td>
</tr>
<tr>
<td>9</td>
<td>RLCALL</td>
<td>44</td>
<td>QOJUL</td>
</tr>
<tr>
<td>10</td>
<td>BXSFCT</td>
<td>45</td>
<td>OFX</td>
</tr>
<tr>
<td>11</td>
<td>COMM</td>
<td>46</td>
<td>QEXP</td>
</tr>
<tr>
<td>12</td>
<td>CASSFCT</td>
<td>47</td>
<td>QINTRP</td>
</tr>
<tr>
<td>13</td>
<td>DENTS</td>
<td>48</td>
<td>OLJ</td>
</tr>
<tr>
<td>14</td>
<td>DSMSC1</td>
<td>49</td>
<td>OMIX</td>
</tr>
<tr>
<td>15</td>
<td>DUMP</td>
<td>50</td>
<td>OPEPP</td>
</tr>
<tr>
<td>16</td>
<td>DUMFX</td>
<td>51</td>
<td>OSAME</td>
</tr>
<tr>
<td>17</td>
<td>ELME</td>
<td>52</td>
<td>OTAO</td>
</tr>
<tr>
<td>18</td>
<td>ELTIME</td>
<td>53</td>
<td>P11</td>
</tr>
<tr>
<td>19</td>
<td>ERPR</td>
<td>54</td>
<td>P12</td>
</tr>
<tr>
<td>20</td>
<td>FGICAL</td>
<td>55</td>
<td>P13</td>
</tr>
<tr>
<td>21</td>
<td>EXACT</td>
<td>56</td>
<td>P14</td>
</tr>
<tr>
<td>22</td>
<td>FINXX</td>
<td>57</td>
<td>RADIUS</td>
</tr>
<tr>
<td>23</td>
<td>FORT7Q</td>
<td>58</td>
<td>READ</td>
</tr>
<tr>
<td>24</td>
<td>GEOM</td>
<td>59</td>
<td>RESTMP</td>
</tr>
<tr>
<td>25</td>
<td>GOMAR</td>
<td>60</td>
<td>RXKT</td>
</tr>
<tr>
<td>26</td>
<td>INGAS</td>
<td>61</td>
<td>SHOCK</td>
</tr>
<tr>
<td>27</td>
<td>INIT</td>
<td>62</td>
<td>SIMO</td>
</tr>
<tr>
<td>28</td>
<td>KANDMU</td>
<td>63</td>
<td>STUNTS</td>
</tr>
<tr>
<td>29</td>
<td>KINT</td>
<td>64</td>
<td>THERM</td>
</tr>
<tr>
<td>30</td>
<td>LIST</td>
<td>65</td>
<td>THROAT</td>
</tr>
<tr>
<td>31</td>
<td>MATINV</td>
<td>66</td>
<td>TRANSP</td>
</tr>
<tr>
<td>32</td>
<td>MODEL</td>
<td>67</td>
<td>WEDGE</td>
</tr>
<tr>
<td>33</td>
<td>NFWRAP</td>
<td>68</td>
<td>WFSGLN</td>
</tr>
<tr>
<td>34</td>
<td>NFXTMP</td>
<td>69</td>
<td>XSFCT</td>
</tr>
<tr>
<td>35</td>
<td>NONE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td></td>
<td>1, 16, 24,</td>
<td></td>
</tr>
<tr>
<td>AA</td>
<td></td>
<td>20, 33, 35, 39,</td>
<td></td>
</tr>
<tr>
<td>AAA</td>
<td></td>
<td>21,</td>
<td></td>
</tr>
<tr>
<td>AAMS</td>
<td>READAT</td>
<td>2, 58,</td>
<td></td>
</tr>
<tr>
<td>ACOM</td>
<td></td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>AE</td>
<td></td>
<td>8,</td>
<td></td>
</tr>
<tr>
<td>AFnts</td>
<td></td>
<td>1, 5, 12, 35, 42,</td>
<td></td>
</tr>
<tr>
<td>AFnx</td>
<td></td>
<td>11, 16, 23, 24, 35, 38, 60, 65,</td>
<td></td>
</tr>
<tr>
<td>AIN</td>
<td>INGNE</td>
<td>2, 16, 25, 35,</td>
<td></td>
</tr>
<tr>
<td>ALGJ</td>
<td></td>
<td>11,</td>
<td></td>
</tr>
<tr>
<td>AM</td>
<td>OUTPUT</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>AMACH</td>
<td></td>
<td>16, 21, 23, 32, 39, 42, 65,</td>
<td></td>
</tr>
<tr>
<td>AMP</td>
<td>BLNE</td>
<td>8, 16,</td>
<td></td>
</tr>
<tr>
<td>AMPR</td>
<td>BLNF</td>
<td>8, 16,</td>
<td></td>
</tr>
<tr>
<td>ANGLE</td>
<td>ROWEDG</td>
<td>2, 30, 38, 67,</td>
<td></td>
</tr>
<tr>
<td>AR</td>
<td></td>
<td>16, 23, 35,</td>
<td></td>
</tr>
<tr>
<td>AREHA</td>
<td></td>
<td>16, 27, 35,</td>
<td></td>
</tr>
<tr>
<td>ARBD</td>
<td></td>
<td>16, 27, 35,</td>
<td></td>
</tr>
<tr>
<td>ASTAR</td>
<td>TRANS4</td>
<td>50, 58,</td>
<td></td>
</tr>
<tr>
<td>ATP1</td>
<td>AREA</td>
<td>16, 25, 58,</td>
<td></td>
</tr>
<tr>
<td>AXIMOD</td>
<td>READAT</td>
<td>2, 58,</td>
<td></td>
</tr>
<tr>
<td>AXISYM</td>
<td>ROWEDG</td>
<td>23, 30, 32, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>AZ</td>
<td></td>
<td>28, 29, 66,</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>TRANS3</td>
<td>28, 29, 66,</td>
<td></td>
</tr>
<tr>
<td>BE</td>
<td></td>
<td>16, 35, 39,</td>
<td></td>
</tr>
<tr>
<td>BFT</td>
<td></td>
<td>16, 20, 21, 26, 33,</td>
<td></td>
</tr>
<tr>
<td>BETA</td>
<td></td>
<td>11, 21, 35, 39,</td>
<td></td>
</tr>
<tr>
<td>BLK</td>
<td></td>
<td>16, 35, 39,</td>
<td></td>
</tr>
<tr>
<td>BLINT</td>
<td>BLNE</td>
<td>8, 16, 35,</td>
<td></td>
</tr>
<tr>
<td>BLNFAR</td>
<td></td>
<td>35,</td>
<td></td>
</tr>
<tr>
<td>BLVAR</td>
<td>ANJIT</td>
<td>23,</td>
<td></td>
</tr>
<tr>
<td>BPR</td>
<td></td>
<td>11, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>BSTAR</td>
<td>TRANS3</td>
<td>28,</td>
<td></td>
</tr>
<tr>
<td>BTA</td>
<td></td>
<td>50, 58,</td>
<td></td>
</tr>
<tr>
<td>BZERO</td>
<td></td>
<td>16, 20, 33, 41, 58,</td>
<td></td>
</tr>
<tr>
<td>BZEROI</td>
<td>READAT</td>
<td>2, 58,</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>1, 16, 24, 66,</td>
<td></td>
</tr>
<tr>
<td>CAI</td>
<td></td>
<td>11, 16, 30, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>CAPO</td>
<td></td>
<td>20, 30, 58,</td>
<td></td>
</tr>
<tr>
<td>CAPX</td>
<td></td>
<td>16, 26, 33, 36, 42,</td>
<td></td>
</tr>
<tr>
<td>CARXTH</td>
<td></td>
<td>16, 36,</td>
<td></td>
</tr>
<tr>
<td>CARB</td>
<td></td>
<td>16, 35,</td>
<td></td>
</tr>
<tr>
<td>CAS</td>
<td></td>
<td>8, 23, 42,</td>
<td></td>
</tr>
<tr>
<td>CATFAC</td>
<td>MODAR</td>
<td>2, 16, 30, 32, 58,</td>
<td></td>
</tr>
<tr>
<td>CHI</td>
<td>RENF</td>
<td>2, 30, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>CCI</td>
<td></td>
<td>26,</td>
<td></td>
</tr>
<tr>
<td>CDPJ</td>
<td></td>
<td>1, 8, 11, 13, 16, 21, 38, 39, 63, 64, 66,</td>
<td></td>
</tr>
<tr>
<td>CDIJ</td>
<td></td>
<td>20, 21, 26, 33, 35,</td>
<td></td>
</tr>
<tr>
<td>CEACT</td>
<td></td>
<td>11, 16, 30, 58,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>EP1</td>
<td>/ELEM  /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>EP10</td>
<td>/ELEM  /</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EP2</td>
<td>/ELEM  /</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EP3</td>
<td>/ELEM  /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>EP4</td>
<td>/ELEM  /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>EP5</td>
<td>/ELEM  /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>EP6</td>
<td>/ELEM  /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>EP7</td>
<td>/ELEM  /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>EP8</td>
<td>/ELEM  /</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ERR</td>
<td>/ERROR /</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTAI</td>
<td>/ERROR /</td>
<td>1, 4, 5, 8, 11, 13, 15, 20, 21, 22, 23, 24, 26, 32, 33, 34, 35, 36, 39, 39, 42, 59, 60, 65, 66, 69,</td>
<td></td>
</tr>
<tr>
<td>ETAJ</td>
<td>/ERROR /</td>
<td>11, 16, 30, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>FACMP</td>
<td>/NEWMP /</td>
<td>16, 27, 30, 58, 67, 64,</td>
<td></td>
</tr>
<tr>
<td>FACNAM</td>
<td>/RROUT /</td>
<td>38, 58,</td>
<td></td>
</tr>
<tr>
<td>FAILED</td>
<td>/ERROR /</td>
<td>11, 13, 16, 35, 60,</td>
<td></td>
</tr>
<tr>
<td>FLEWIS</td>
<td>/TAROP /</td>
<td>1, 16, 32, 66,</td>
<td></td>
</tr>
<tr>
<td>FLOW</td>
<td>/RROUT /</td>
<td>1, 38, 58,</td>
<td></td>
</tr>
<tr>
<td>FLUX</td>
<td>/RROUT /</td>
<td>1, 16, 23, 33, 35, 36, 41,</td>
<td></td>
</tr>
<tr>
<td>FSTAG</td>
<td>/MONPAR/</td>
<td>2, 16, 30, 32, 58,</td>
<td></td>
</tr>
<tr>
<td>FOUT</td>
<td>/OUTPUT /</td>
<td>38, 67,</td>
<td></td>
</tr>
<tr>
<td>GAMIN</td>
<td>/RENF /</td>
<td>2, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>GAMMA</td>
<td>/OUTPUT /</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>GFLJ</td>
<td>/OUTPUT /</td>
<td>30, 59, 64,</td>
<td></td>
</tr>
<tr>
<td>GJ</td>
<td>/OUTPUT /</td>
<td>1, 8, 9, 11, 21, 23, 32, 33, 35, 36, 38, 39, 60,</td>
<td></td>
</tr>
<tr>
<td>GJA</td>
<td>/OUTPUT /</td>
<td>1, 16, 23, 26, 35, 36, 41, 63,</td>
<td></td>
</tr>
<tr>
<td>GJN</td>
<td>/OUTPUT /</td>
<td>16, 35, 60,</td>
<td></td>
</tr>
<tr>
<td>GJMF</td>
<td>/OUTPUT /</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>GP</td>
<td>/MIXT /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>GP1</td>
<td>/MIXT /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>GP2</td>
<td>/MIXT /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>GP3</td>
<td>/MIXT /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>GP4</td>
<td>/MIXT /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>GP5</td>
<td>/MIXT /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>GP6</td>
<td>/MIXT /</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>GSO</td>
<td>/RENF /</td>
<td>35,</td>
<td></td>
</tr>
<tr>
<td>GTFST</td>
<td>/RENF /</td>
<td>2, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>/RENT /</td>
<td>8, 38, 50,</td>
<td></td>
</tr>
<tr>
<td>HDFLX</td>
<td>/RENT /</td>
<td>16, 35, 60,</td>
<td></td>
</tr>
<tr>
<td>H+</td>
<td>/RENT /</td>
<td>30, 32, 35, 38, 58, 63,</td>
<td></td>
</tr>
<tr>
<td>HR</td>
<td>/RENT /</td>
<td>2, 8, 16, 38,</td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>/RENT /</td>
<td>33, 38, 50, 59,</td>
<td></td>
</tr>
<tr>
<td>HSTAG</td>
<td>/READAT/</td>
<td>86,</td>
<td></td>
</tr>
<tr>
<td>HTEST</td>
<td>/RENE /</td>
<td>2, 35, 69,</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>/TRANS2/</td>
<td>7, 14, 63, 69,</td>
<td></td>
</tr>
<tr>
<td>IAMDIS</td>
<td>/ROMED /</td>
<td>23,</td>
<td></td>
</tr>
<tr>
<td>IATOM</td>
<td>/LINT /</td>
<td>69,</td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>/LINT /</td>
<td>16, 32, 33, 38, 58,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>ISSP3</td>
<td>/</td>
<td>16, 21, 35, 39, 60,</td>
<td></td>
</tr>
<tr>
<td>ISSP4</td>
<td>/</td>
<td>16, 21, 35, 69,</td>
<td></td>
</tr>
<tr>
<td>ISSP5</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP6</td>
<td>/</td>
<td>16, 16, 58, 63, 66,</td>
<td></td>
</tr>
<tr>
<td>ISSP7</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP8</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP9</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP10</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP11</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP12</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP13</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP14</td>
<td>/</td>
<td>16, 16, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP15</td>
<td>/</td>
<td>READAT/3, 43, 47, 58, 66,</td>
<td></td>
</tr>
<tr>
<td>ISSP16</td>
<td>/</td>
<td>ROCTR/2, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP17</td>
<td>/</td>
<td>RWEDG/2, 58, 67,</td>
<td></td>
</tr>
<tr>
<td>ISSP18</td>
<td>/</td>
<td>16, 35,</td>
<td></td>
</tr>
<tr>
<td>ISSP19</td>
<td>/</td>
<td>TAPOUT/1, 2, 16, 32, 38, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP20</td>
<td>/</td>
<td>TAPOUT/1, 2, 16, 32, 38, 58,</td>
<td></td>
</tr>
<tr>
<td>ISSP21</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP22</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP23</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP24</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP25</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP26</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP27</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP28</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP29</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP30</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>ISSP31</td>
<td>/</td>
<td>TAPOUT/1, 32, 48,</td>
<td></td>
</tr>
<tr>
<td>I355</td>
<td>/</td>
<td>16, 35, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>110</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>112</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>113</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>114</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>116</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>117</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>118</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>119</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>122</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>123</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>124</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>125</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>126</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>127</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>129</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>130</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>131</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>2355</td>
<td>/</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>/TRANI</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINE</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>J26</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J27</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J28</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J29</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J30</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J31</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J32</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J33</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J34</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J35</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J36</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J37</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J38</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J39</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J40</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J41</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J42</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J43</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J44</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J45</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J46</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J47</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J48</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J49</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J50</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J51</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J52</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J53</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J54</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J55</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J56</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J57</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J58</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J59</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J60</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J61</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J62</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J63</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J64</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J65</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J66</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J67</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J68</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J69</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>J70</td>
<td>TRANJ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>TABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>J71</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J72</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J73</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J74</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J75</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J76</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J77</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J78</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J79</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J8</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J80</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J81</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J82</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J83</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J84</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J85</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J86</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J87</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J88</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J89</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J9</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J90</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J91</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J92</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J93</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J94</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J95</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J96</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J97</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J98</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>J99</td>
<td>/TRANJ /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>KDIM</td>
<td>/MODPAR/</td>
<td>2, 16, 30, 32, 53,</td>
<td></td>
</tr>
<tr>
<td>KQO</td>
<td>/TRANS2/</td>
<td>3, 58, 63, 69,</td>
<td></td>
</tr>
<tr>
<td>KKQ</td>
<td>/TRANS2/</td>
<td>10, 43, 63, 69,</td>
<td></td>
</tr>
<tr>
<td>KT</td>
<td>/TRANS7/</td>
<td>11, 16, 30, 59,</td>
<td></td>
</tr>
<tr>
<td>KTB</td>
<td>/TNSNFO/</td>
<td>11, 16, 30, 59,</td>
<td></td>
</tr>
<tr>
<td>KUR</td>
<td>/</td>
<td>11, 30, 58,</td>
<td></td>
</tr>
<tr>
<td>LC</td>
<td>/</td>
<td>16, 35,</td>
<td></td>
</tr>
<tr>
<td>LC</td>
<td>/</td>
<td>2, 30, 32, 59,</td>
<td></td>
</tr>
<tr>
<td>LIMOUT</td>
<td>/ROUT /</td>
<td>2, 30, 32, 59,</td>
<td></td>
</tr>
<tr>
<td>LDIJ</td>
<td>/</td>
<td>21, 26, 30, 32, 35, 39, 58, 69,</td>
<td></td>
</tr>
<tr>
<td>LSS</td>
<td>/TRANS5/</td>
<td>69,</td>
<td></td>
</tr>
<tr>
<td>MBL</td>
<td>/MDPT /</td>
<td>24, 58,</td>
<td></td>
</tr>
<tr>
<td>VET</td>
<td>/VMAIN/</td>
<td>1, 58,</td>
<td></td>
</tr>
<tr>
<td>MIDPT</td>
<td>/MODPT2</td>
<td>2, 30, 32, 59,</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>/</td>
<td>1, 35, 42,</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>/TRANS2/</td>
<td>16, 26, 33, 36,</td>
<td></td>
</tr>
<tr>
<td>NANGLE</td>
<td>/ROWFDS/</td>
<td>2, 30, 59, 67,</td>
<td></td>
</tr>
<tr>
<td>NPL</td>
<td>/AREA /</td>
<td>2, 16, 30, 58,</td>
<td></td>
</tr>
<tr>
<td>NCS</td>
<td>/COLDSP/</td>
<td>1, 2, 32, 59,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>PRESAI</td>
<td>READAT</td>
<td>50, 23</td>
<td></td>
</tr>
<tr>
<td>PRESB</td>
<td></td>
<td>16, 23</td>
<td></td>
</tr>
<tr>
<td>PRESTH</td>
<td></td>
<td>1, 16, 23, 36</td>
<td></td>
</tr>
<tr>
<td>PRE</td>
<td>TRPROP</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>PR</td>
<td></td>
<td>9, 16, 35, 39, 42</td>
<td></td>
</tr>
<tr>
<td>PR1</td>
<td>RSPRP</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>PR2</td>
<td>GLOUT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PR3</td>
<td>TRANS1</td>
<td>2, 8, 16, 38, 43</td>
<td></td>
</tr>
<tr>
<td>O1</td>
<td>TNCF</td>
<td>7, 16, 21, 39, 51, 54, 55, 56, 63, 66, 69</td>
<td></td>
</tr>
<tr>
<td>O2</td>
<td>TNCF</td>
<td>11, 16, 21, 39, 51, 54, 55, 56, 63, 66, 69</td>
<td></td>
</tr>
<tr>
<td>O3</td>
<td></td>
<td>11, 16, 21, 39, 51, 54, 55, 56, 63, 66, 69</td>
<td></td>
</tr>
<tr>
<td>O4</td>
<td>TNCF</td>
<td>11, 16, 21, 39, 51, 54, 55, 56, 63, 66, 69</td>
<td></td>
</tr>
<tr>
<td>O5</td>
<td>GLOUT</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>O6</td>
<td>GLOUT</td>
<td>16, 20, 26, 33</td>
<td></td>
</tr>
<tr>
<td>OPJ</td>
<td>COLDSP</td>
<td>1, 8, 12, 59</td>
<td></td>
</tr>
<tr>
<td>OQ</td>
<td></td>
<td>11, 16, 20, 39, 58</td>
<td></td>
</tr>
<tr>
<td>OQ1</td>
<td></td>
<td>2, 16, 39</td>
<td></td>
</tr>
<tr>
<td>OQ2</td>
<td>RENE</td>
<td>2, 16, 39</td>
<td></td>
</tr>
<tr>
<td>OQ3</td>
<td>GLOUT</td>
<td>2, 8, 16, 39</td>
<td></td>
</tr>
<tr>
<td>OQ4</td>
<td>ROWEDG</td>
<td>2, 30, 55, 67</td>
<td></td>
</tr>
<tr>
<td>OQ5</td>
<td>GREADAT</td>
<td>2, 59</td>
<td></td>
</tr>
<tr>
<td>OQ6</td>
<td>GREADAT</td>
<td>2, 59</td>
<td></td>
</tr>
<tr>
<td>OQ7</td>
<td></td>
<td>2, 8, 16, 39</td>
<td></td>
</tr>
<tr>
<td>OQ8</td>
<td></td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>OQ9</td>
<td></td>
<td>1, 8, 11, 16, 21, 26, 32, 38, 39, 59</td>
<td></td>
</tr>
<tr>
<td>OQ10</td>
<td></td>
<td>41, 42, 60</td>
<td></td>
</tr>
<tr>
<td>OQ11</td>
<td>RMC</td>
<td>16, 23, 35</td>
<td></td>
</tr>
<tr>
<td>OQ12</td>
<td>RMCBAR</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>OQ13</td>
<td>RMC</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>OQ14</td>
<td>RMCINF</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>OQ15</td>
<td>OUTPUT</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>OQ16</td>
<td></td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>OQ17</td>
<td></td>
<td>11, 16, 21</td>
<td></td>
</tr>
<tr>
<td>R1</td>
<td></td>
<td>11, 16</td>
<td></td>
</tr>
<tr>
<td>R1H</td>
<td></td>
<td>11, 16</td>
<td></td>
</tr>
<tr>
<td>R1H1</td>
<td></td>
<td>11, 16, 23, 24, 36</td>
<td></td>
</tr>
<tr>
<td>R1H2</td>
<td></td>
<td>11, 16, 23, 26, 36, 38</td>
<td></td>
</tr>
<tr>
<td>R1PARA</td>
<td></td>
<td>11, 16, 23, 26, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R1PARP</td>
<td></td>
<td>11, 16, 23, 26, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R2</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R2P</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R3</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R4</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R5</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R7</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R8</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R9</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R10</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R11</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R12</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R13</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R14</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R15</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R16</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R17</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R18</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R19</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R20</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R21</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R22</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R23</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R24</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R25</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R26</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R27</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R28</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R29</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>R30</td>
<td>REAC</td>
<td>16, 23, 36, 41</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING Routines</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>RP70</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP71</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP72</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP73</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP74</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP75</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP76</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP77</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP78</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP79</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP80</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP81</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP82</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP83</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP84</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP85</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP86</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP87</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP88</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP89</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP90</td>
<td>RFAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP91</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>RP92</td>
<td>REAC</td>
<td>2, 58</td>
<td></td>
</tr>
<tr>
<td>R9A</td>
<td>THRT</td>
<td>16, 25, 35, 46, 59, 65</td>
<td></td>
</tr>
<tr>
<td>RT61</td>
<td>AREA</td>
<td>1, 3, 4, 5, 8, 14, 24, 25, 38, 58</td>
<td></td>
</tr>
<tr>
<td>R0</td>
<td>AL</td>
<td>1, 4, 5, 8, 16, 24, 38, 57</td>
<td></td>
</tr>
<tr>
<td>SAJ</td>
<td></td>
<td>16, 27, 64</td>
<td></td>
</tr>
<tr>
<td>SAVF</td>
<td>TEMPY</td>
<td>16, 27, 64</td>
<td></td>
</tr>
<tr>
<td>SHJ</td>
<td></td>
<td>1, 8, 23, 32, 38, 63, 66</td>
<td></td>
</tr>
<tr>
<td>SHJ</td>
<td></td>
<td>30, 58</td>
<td></td>
</tr>
<tr>
<td>SC</td>
<td></td>
<td>16, 35</td>
<td></td>
</tr>
<tr>
<td>SCPG</td>
<td></td>
<td>8, 11, 16, 21</td>
<td></td>
</tr>
<tr>
<td>SCPCH</td>
<td>TNCE</td>
<td>8, 11, 16, 21</td>
<td></td>
</tr>
<tr>
<td>SCHA</td>
<td>TNERK</td>
<td>35, 60</td>
<td></td>
</tr>
<tr>
<td>SDCHI</td>
<td></td>
<td>16, 35, 39, 42</td>
<td></td>
</tr>
<tr>
<td>SDCJ</td>
<td></td>
<td>35, 60</td>
<td></td>
</tr>
<tr>
<td>SDG</td>
<td></td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>SDJ</td>
<td></td>
<td>16, 35, 60</td>
<td></td>
</tr>
<tr>
<td>SOT</td>
<td></td>
<td>16, 35, 60</td>
<td></td>
</tr>
<tr>
<td>SOTF</td>
<td>TNERK</td>
<td>16, 35, 60</td>
<td></td>
</tr>
<tr>
<td>SF1</td>
<td></td>
<td>1, 11, 16, 26, 35, 38</td>
<td></td>
</tr>
<tr>
<td>SFN</td>
<td></td>
<td>11, 13, 16, 20, 32, 33, 39, 41, 63, 64</td>
<td></td>
</tr>
<tr>
<td>S1J</td>
<td></td>
<td>11, 13, 16, 20, 32, 33, 39, 41, 63, 64</td>
<td></td>
</tr>
<tr>
<td>SHJAP</td>
<td></td>
<td>16, 27, 32, 63, 64</td>
<td></td>
</tr>
<tr>
<td>SHJDGJ</td>
<td>DRV</td>
<td>8, 16, 21</td>
<td></td>
</tr>
<tr>
<td>SHDP</td>
<td></td>
<td>11, 16, 41</td>
<td></td>
</tr>
<tr>
<td>SHV</td>
<td></td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>SIGMA</td>
<td>TPRPS</td>
<td>1, 16, 23, 32, 39, 66</td>
<td></td>
</tr>
<tr>
<td>SIGP</td>
<td>RESPRP</td>
<td>1, 23</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>SKIL</td>
<td></td>
<td>16, 20, 33</td>
<td></td>
</tr>
<tr>
<td>SL</td>
<td></td>
<td>16, 35, 58</td>
<td></td>
</tr>
<tr>
<td>SL64</td>
<td></td>
<td>11, 16, 35</td>
<td></td>
</tr>
<tr>
<td>SM</td>
<td></td>
<td>1, 16, 23, 24, 35, 36, 38, 59</td>
<td></td>
</tr>
<tr>
<td>S'MASS</td>
<td>MSSFL</td>
<td>1, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S'N</td>
<td>/HOUT</td>
<td>2, 8, 16, 38</td>
<td></td>
</tr>
<tr>
<td>SPHP</td>
<td>/SPEC</td>
<td>27, 59, 63</td>
<td></td>
</tr>
<tr>
<td>S1'1</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'10</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'11</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'12</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'13</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'14</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'15</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'16</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'17</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'18</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'19</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'20</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'21</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'22</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'23</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'24</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'25</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'26</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'27</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'28</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'29</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>S1'30</td>
<td>/SPEC</td>
<td>2, 59, 59</td>
<td></td>
</tr>
<tr>
<td>SRTA</td>
<td>/AEGFOM</td>
<td>4, 16, 24</td>
<td></td>
</tr>
<tr>
<td>S5</td>
<td></td>
<td>16, 23</td>
<td></td>
</tr>
<tr>
<td>SS</td>
<td></td>
<td>11, 16, 21, 23, 24, 32, 33, 38, 41, 42, 60</td>
<td></td>
</tr>
<tr>
<td>SUG</td>
<td></td>
<td>1, 16, 21</td>
<td></td>
</tr>
<tr>
<td>SUNW</td>
<td>/TNCE</td>
<td>11, 16, 21</td>
<td></td>
</tr>
<tr>
<td>SUGN</td>
<td>/RDMAIN</td>
<td>1, 30, 59</td>
<td></td>
</tr>
<tr>
<td>SUGQ</td>
<td>/RDMAIN</td>
<td>1, 30, 59</td>
<td></td>
</tr>
<tr>
<td>SU2</td>
<td></td>
<td>11, 16, 21, 33, 39, 41</td>
<td></td>
</tr>
<tr>
<td>SK</td>
<td>/RL</td>
<td>1, 8, 15</td>
<td></td>
</tr>
<tr>
<td>S1</td>
<td>/AEGFOM</td>
<td>4, 16, 24, 65</td>
<td></td>
</tr>
<tr>
<td>S2</td>
<td>/AFGFOH</td>
<td>4, 16, 24, 65</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>/TRANS1</td>
<td>44, 46, 46, 48, 50, 52, 53, 66, 69</td>
<td></td>
</tr>
<tr>
<td>TAUW</td>
<td>/RIOUT</td>
<td>2, 8, 15, 38</td>
<td></td>
</tr>
<tr>
<td>TO</td>
<td></td>
<td>16, 35</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>T97</td>
<td>OUTMOD</td>
<td>32, 38,</td>
<td></td>
</tr>
<tr>
<td>TEST</td>
<td>/</td>
<td>16, 20, 27, 33,</td>
<td></td>
</tr>
<tr>
<td>TESTB</td>
<td>/</td>
<td>16, 23, 27, 36,</td>
<td></td>
</tr>
<tr>
<td>TE TEST</td>
<td>RENE</td>
<td>2, 35, 59,</td>
<td></td>
</tr>
<tr>
<td>TFA</td>
<td>/</td>
<td>16, 30, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>TFB</td>
<td>/</td>
<td>16, 27, 30, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>TFC</td>
<td>/</td>
<td>16, 27, 30, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>TFD</td>
<td>/</td>
<td>16, 27, 30, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>TFE</td>
<td>/</td>
<td>16, 30, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>THEGA</td>
<td>/BLOUT</td>
<td>2, 8, 16, 38,</td>
<td></td>
</tr>
<tr>
<td>THEV</td>
<td>/</td>
<td>16, 27, 64,</td>
<td></td>
</tr>
<tr>
<td>THEVE</td>
<td>POLYAT</td>
<td>27, 30, 59, 64,</td>
<td></td>
</tr>
<tr>
<td>THE VP</td>
<td>/</td>
<td>30, 58,</td>
<td></td>
</tr>
<tr>
<td>TINF</td>
<td>/OUTPUT /</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>TL</td>
<td>/TRANS 4</td>
<td>47, 58,</td>
<td></td>
</tr>
<tr>
<td>TL 1</td>
<td>/TRANS 4</td>
<td>40, 58, 3,</td>
<td></td>
</tr>
<tr>
<td>TL 10</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 11</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 12</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 13</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 14</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 15</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 16</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 17</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 18</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 19</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 20</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 21</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 22</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 23</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 24</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 25</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 26</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 27</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 28</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 29</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 30</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 31</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 32</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 33</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 34</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 35</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 36</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 37</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 38</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL 39</td>
<td>/TRANS 4</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td>TL4</td>
<td>TRANS4/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL4O</td>
<td>TRANS4/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TLS</td>
<td>TRANS4/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL4</td>
<td>TRANS4/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL7</td>
<td>TRANS4/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TL8</td>
<td>TRANS4/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>TMODEL</td>
<td>/MODPAR/</td>
<td>2, 16, 32, 32, 59,</td>
<td></td>
</tr>
<tr>
<td>TNP</td>
<td>/TNEF/</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>TNP1</td>
<td>/TNEF/</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>TNP2</td>
<td>/TNEF/</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>TPLATE</td>
<td>/MODPAR/</td>
<td>2, 30, 58,</td>
<td></td>
</tr>
<tr>
<td>TPRINT</td>
<td>/READAT/</td>
<td>2, 58,</td>
<td></td>
</tr>
<tr>
<td>TPROP</td>
<td>/TPROP/</td>
<td>63,</td>
<td></td>
</tr>
<tr>
<td>TPS</td>
<td>/TPROP/</td>
<td>2,</td>
<td></td>
</tr>
<tr>
<td>TS</td>
<td>/STAG/</td>
<td>16, 32,</td>
<td></td>
</tr>
<tr>
<td>TSAR</td>
<td>/MODPT/</td>
<td>16, 34, 58,</td>
<td></td>
</tr>
<tr>
<td>TSDIAM</td>
<td>/MODPT/</td>
<td>2, 16, 30, 59,</td>
<td></td>
</tr>
<tr>
<td>TSF</td>
<td>/STAG/</td>
<td>16,</td>
<td></td>
</tr>
<tr>
<td>TSTOP</td>
<td>/MASHFL/</td>
<td>2, 17, 58,</td>
<td></td>
</tr>
<tr>
<td>TT</td>
<td>/TSTEP/</td>
<td>63,</td>
<td></td>
</tr>
<tr>
<td>TSTEP</td>
<td>/STAG/</td>
<td>2, 16, 23, 27, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>TWALL</td>
<td>/BL/</td>
<td>1, 2, 3, 16, 19, 58,</td>
<td></td>
</tr>
<tr>
<td>TWEDGE</td>
<td>/TWEDEG/</td>
<td>2, 30, 58, 67,</td>
<td></td>
</tr>
<tr>
<td>T2E</td>
<td>/STAG/</td>
<td>23, 32, 35,</td>
<td></td>
</tr>
<tr>
<td>T2F</td>
<td>/STAG/</td>
<td>23, 32, 35,</td>
<td></td>
</tr>
<tr>
<td>U</td>
<td>/OUTPUT/</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>UINF</td>
<td>/OUTPUT/</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>UP</td>
<td>/OUTPUT/</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>US</td>
<td>/SS/</td>
<td>1, 8, 21, 23, 42,</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>/TRANS7/</td>
<td>10, 43, 53, 69,</td>
<td></td>
</tr>
<tr>
<td>VISCR</td>
<td>/TPROP/</td>
<td>1, 8, 16, 23, 32, 39, 66, 67,</td>
<td></td>
</tr>
<tr>
<td>VISR0T</td>
<td>/BL/</td>
<td>1, 16,</td>
<td></td>
</tr>
<tr>
<td>VV4UNF</td>
<td>/OUTPUT/</td>
<td>67,</td>
<td></td>
</tr>
<tr>
<td>VV</td>
<td>/TRANS/</td>
<td>63, 69,</td>
<td></td>
</tr>
<tr>
<td>VV1</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV10</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV11</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV12</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV13</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV14</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV15</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV16</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV17</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV18</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VV19</td>
<td>/TRANS/</td>
<td>3,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>VV20</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV21</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV22</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV23</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV24</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV25</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV26</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV27</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV28</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV29</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV30</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV31</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV32</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV33</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV34</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV35</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV36</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV37</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV38</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV39</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV40</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV41</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV42</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV43</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV44</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV45</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV46</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV47</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV48</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV49</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV50</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV51</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV52</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV53</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV54</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV55</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV56</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV57</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV58</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV59</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV60</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV61</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV62</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV63</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VV64</td>
<td>/TRANV</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>---------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>VV65</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV66</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV67</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV68</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV69</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV70</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV71</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV72</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV73</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV74</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV75</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV76</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV77</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV78</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV79</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV80</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV91</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV92</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV93</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV94</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV95</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV96</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV97</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV98</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VV99</td>
<td>TRANV</td>
<td></td>
<td></td>
</tr>
<tr>
<td>V10</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V110</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V11</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V12</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V13</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V14</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V15</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V16</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V17</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V18</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V19</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>V20</td>
<td>TRANV</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>V65</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V66</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V67</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V68</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V69</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V7</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V70</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V71</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V72</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V73</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V74</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V75</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V76</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V77</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V78</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V79</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V8</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V80</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V81</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V82</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V83</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V84</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V85</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V86</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V87</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V88</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V89</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V9</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V90</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V91</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V92</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V93</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V94</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V95</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V96</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V97</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V98</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>V99</td>
<td>TRANV</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>W</td>
<td>SWITCH</td>
<td>16, 23, 35</td>
<td></td>
</tr>
<tr>
<td>WEDGEM</td>
<td>RDWDGD</td>
<td>36, 32, 58</td>
<td></td>
</tr>
<tr>
<td>WX</td>
<td>RDWDGD</td>
<td>2, 30, 58, 67</td>
<td></td>
</tr>
<tr>
<td>WSAVE</td>
<td>AVG</td>
<td>2, 23, 35, 58</td>
<td></td>
</tr>
<tr>
<td>WX1</td>
<td>RDWDGD</td>
<td>2, 30, 58, 67</td>
<td></td>
</tr>
<tr>
<td>WX2</td>
<td>RDWDGD</td>
<td>2, 30, 58, 67</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>/</td>
<td>8, 25, 38, 66</td>
<td></td>
</tr>
<tr>
<td>XI</td>
<td>BLNF</td>
<td>8, 16</td>
<td></td>
</tr>
<tr>
<td>XMJAT</td>
<td>/</td>
<td>13, 16, 20, 33, 63, 64</td>
<td></td>
</tr>
<tr>
<td>XMJATO</td>
<td>TNDC</td>
<td>11, 13, 19</td>
<td></td>
</tr>
<tr>
<td>XMOP1</td>
<td>MODPAR</td>
<td>2, 16, 33, 58</td>
<td></td>
</tr>
<tr>
<td>XMP1</td>
<td>MODPAR</td>
<td>16, 23, 35, 58</td>
<td></td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>XNUJ</td>
<td>/</td>
<td>11, 16, 35</td>
<td></td>
</tr>
<tr>
<td>XNUIJ</td>
<td>/</td>
<td>11, 30, 35, 58</td>
<td></td>
</tr>
<tr>
<td>XNUJJP</td>
<td>/</td>
<td>11, 30, 35, 58</td>
<td></td>
</tr>
<tr>
<td>XE</td>
<td>/SWITCH/</td>
<td>8, 23</td>
<td></td>
</tr>
<tr>
<td>XPR</td>
<td>/SWITCH/</td>
<td>16, 23, 35</td>
<td></td>
</tr>
<tr>
<td>XSN</td>
<td>/BLOUT/</td>
<td>2, 8, 16, 35, 38</td>
<td></td>
</tr>
<tr>
<td>XXX</td>
<td>/TAPOUT/</td>
<td>1, 38</td>
<td></td>
</tr>
<tr>
<td>XZERD</td>
<td>/BL</td>
<td>8, 16, 23, 58</td>
<td></td>
</tr>
<tr>
<td>YZEROJ</td>
<td>/READAT/</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>YOZO</td>
<td>/BLRAD/</td>
<td>8, 57</td>
<td></td>
</tr>
<tr>
<td>ZCAP</td>
<td>/EOC</td>
<td>16, 20, 26, 32</td>
<td></td>
</tr>
<tr>
<td>ZCH</td>
<td>/EOC</td>
<td>16, 20, 26, 32</td>
<td></td>
</tr>
<tr>
<td>ZCH</td>
<td>/EOC</td>
<td>16, 20, 26, 32</td>
<td></td>
</tr>
<tr>
<td>ZGMU</td>
<td>/EOC2</td>
<td>2, 26</td>
<td></td>
</tr>
<tr>
<td>ZM2</td>
<td>/TRANS1</td>
<td>28, 65, 69</td>
<td></td>
</tr>
<tr>
<td>ZP</td>
<td></td>
<td>16, 33, 41</td>
<td></td>
</tr>
<tr>
<td>ZP2</td>
<td></td>
<td>16, 23, 26, 33</td>
<td></td>
</tr>
<tr>
<td>ZP20</td>
<td>/NOZZ</td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>ZP2</td>
<td>/EOC2</td>
<td>2, 26</td>
<td></td>
</tr>
<tr>
<td>ZP1</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP10</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP11</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP12</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP13</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP14</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP15</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP16</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP17</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP18</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP19</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP20</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP20</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP2</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP3</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP4</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP5</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP6</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP7</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP8</td>
<td>/NOZZ</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZP9</td>
<td>/EOC</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ZRHO</td>
<td>/EOC</td>
<td>16, 20, 26</td>
<td></td>
</tr>
<tr>
<td>ZSN</td>
<td>/EOC</td>
<td>16, 20, 26, 32</td>
<td></td>
</tr>
</tbody>
</table>

COMPLETED
6. LISTING OF THE NATA CODE

The present section consists of a complete source program listing of the NATA code. The routines appear in the same order in which their names are listed at the beginning of Section 5, i.e., main program, block data routines, and then all subroutines and functions in alphanumeric order of their names.
MAIN PROGRAM FOR NOZZLE FLOW SOLUTION

LOGICAL SUPGO, DATAVE, NOTRAN
LOGICAL ERR
REAL CT, ETG
REAL AGM(30), ELMEN(10), HP(20)
DOUBLE PRECISION AA, AAA, CAP, CJ, CDJ
COMMON AA(22, 24), CDI(20, 10), CAPX(20)
COMMON AA(22, 24), AFNX, AMACH, AR, ARBA, ARBB, BZERO,
       C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP,
       CRRB, CR, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMX,
       CTP, CTPL, CTT, CX, CXB, CMAX, DAT, DBTEST, DELT1,
       DELT2, DELTAX, DLOGAI, DLOGAR, DLOGAT, DLOGAR,
       DLOGAR, DP, DPRE, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
       DPRET, DRES, DPRET, DRES, DPRET, DRES, DPRET,
       DRES, DPRET, DRES, DPRET, DRES, DPRET, DRES,
COMMON /OUTPUT/ FVOUT(35),GJMF(20)
COMMON /RDOUT/ FLOW,FACNAM,CHANAM,LIMOL
COMMON /SS/ CAS,US
COMMON /SPEC/ SPRT(43,30)
COMMON /LN/ ISATOM,ISMOL,JATOM,JMOL
COMMON /ELEM/ EEP(2,10)
COMMON /THRT/ RSA
DATA NCASE,NERRR,NRCTOT,IRCO /4*0/, ONE /1e/, PIV /3e1A159/

ISW1A=1
ISW2A=1
ISW3A=1
ISW4A=0
ISW5A=0
ISW6A=0
ISW1B=0
ISW2B=0
IS3=1
ISW4B=0
ISW5B=0
ISW6B=1
IRUN=0
MFITER=1
NOTRAN=.FALSE.

10 CALL ELTIME (ETO.0)
CALL READ
IS3=ISW3B
IF (.NOT.*NOTRAN) GO TO 20
ISW3B=0
VISC=0.
PR=0.
SIGMA=0.
FLEWIS=0.

20 CALL LIST
IF (.NOTRAN) WRITE (6,330)
ET=ETO
CALL ELTIME (ET,1)
NRCOUT=0
NCASE=NCASE+1
ERR=.FALSE.
OMDIST(1)=1.
OMDIST(2)=1.
IF (ISW2B.EQ.0) SM=0.
CALL INGAS
IF (ERR) GO TO 290
IF (ISW2B.EQ.0) CALL RESTMP
IF (ERR) GO TO 290
IF (ISW2B.LE.0) GO TO 30
CALL INIT
CALL INTA
IF (ERR) GO TO 290
IF (ISW6A.GE.0) GO TO 40
CALL STUNTS
IRCO=1
GO TO 300
RHO=1.
SU=0.
CH=CH
CM=CM
WRITE (6,370)
CALL OUT1
IF (ERR) GO TO 290
CALL ELTIME (ET,1)
IF (ISW1.EQ.2 OR ISW2.LT.0) GO TO 300
IF (NOTRAN) GO TO 160
CALL TRANSP (CTAP, PRESA)
IF (ERR) GO TO 290
VISCC=VISC*CF(3)
WRITE (6,380) VISC,PR,SIGMA,FLEWIS
CALL OUT1
CALL ELTIME (ET,1)
CALL NRMAX
IF (ERR) GO TO 290
PRESTH=PRES
UTH=SU
CHTH=CH
CMTH=CM
DO 200 I=1,ISS
GMF(I)=GJ(I)*CM
IF (ISW3A.EQ.0) GO TO 220
ISOLN=2
IF (DATAPE) GO TO 170
CALL EQUIL (2)
IF (ERR) GO TO 290
I=M=1
JK=O
ISMCNR=ISMC
ISSNR=ISS
WRITE (6.400)
CT=CTMAX
PRES=PRESTM
RHQ=RHTH
SU=UTH
CH=CHTH
CM=CMTH
DO 230 I=1,ISS
SAVEC(I)=GMF(I)
CALL OUTI
IF (ERR) GO TO 290
CALL ELTIME (ET*1)
IF (ISW2A.EQ.0) GO TO 300
DO 240 I=1,ISS
GJ(I)=GJA(I)
WRITE (6.400)
PRES=100
N=0
SUM1=ALOG(RHTH)
A=4. +5.*SUM1
N=N+1
IF (N-50) 260,270,270
SUM2=A+2.*
SUM3=RHTH**A
D=SUM3*SUM2-2.*
D1=SUM3*(SUM2+SUM1+1.)
ATEST=A
A=A-D/D1
IF (ABS(D) GT .00001) GO TO 250
IF (ABS(ATEST-A) GT .00001) GO TO 250
C=(RHTH**2)*(1.-RHTH**A)
DELT=DELT1
IF (ISW3B.EQ.0) DELT=0.049*(1.-CTMAX)
CT=1.0-DELT
CM=CMA
CALL NEWRAP
IF (ERR) GO TO 290
AFNTS=SM/FLUX
CALL FINDX (AFNTS,ONE,CX)
WRITE (6.340) A,C
ISOLN=3
IF (DATAPE) GO TO 170
CALL NONEQ
CALL ELTIME (ET*1)
IF (.NOT.ERR) GO TO 300
CALL DUMPEx
NERROR=NERROR+1
NRCTOT=NRCTOT+NRCOUT
WRITE (6,420) IRUN,NCASE,NRCOUT,ITPOUT,NRCTOT
ELTHIN=(ET-ETO)/60.
WRITE (6,410) ELTHS
IF (ISW4A) 10,310,10
310 NCOMPL=NCASE-NERROR
WRITE (6,350) NCOMPL,NERROR
IF (*NOT*DATAPE) GO TO 320
END FILE ITPOUT
REWIND ITPOUT
CALL EXIT

FORMAT (43HTRANSPORT PROPERTY CALCULATIONS SUPPRESSED) MAI 292
340 FORMAT (20H DENSITY F1T-ALPHA=,1PE15.7,10H CONSTANT=,E15.7) MAI 293
350 FORMAT (///15.16H CASES COMPLETED/15.13H CASES FAILED) MAI 294
360 FORMAT (27H SPECIFIC HEAT OF COLD GAS=F7.4,16H BTU/LB-DEG R AT,F5,2MAI 295
1.2*6 DEG K) MAI 296
370 FORMAT (1H1.9X,24H- RESERVOIR CONDITIONS -) MAI 297
380 FORMAT (///31H RESERVOIR TRANSPORT PROPERTIES//19H VISCOSITY MAI 298
1.1=3X,1PE9.2,11H LBMI/FT-SEC/19H PRANDTL NUMBER =.0PF125/19H SIMILMAI 299
2=3X,1PE9.2,7H MHO/CM/19H LEWIS NUMBER =.0PF10.19H LEWIS NUMBER MAI 300
331 ) MAI 301
390 FORMAT (1H1.7X,28H- FROZEN THROAT CONDITIONS -) MAI 302
400 FORMAT (1H1.6X,33H- EQUILIBRIUM THROAT CONDITIONS -) MAI 303
410 FORMAT (24H CASE EXECUTION TIME WAS,F6.2,8H MINUTES) MAI 304
420 FORMAT (8HORUN NO.,17.5X,4CASE,14.23H OF THIS JOB COMPLETED.,15.2MAI 305
14H RECORDS WRITTEN ON TAPE,13.1H,17.36H RECORDS WRITTEN SO FAR INMAI 306
2 THIS JOB,) MAI 307
END
BLOCK DATA
LOGICAL DATAE, WEGDM, AXI SYM, READXS, RE, AAMS, AXIMOD
DOUBLE PRECISION AIN
   EP8(2), EP9(2), EP10(2)
COMMON /SPEC/ ADAMS, EP2(4), SP2(4), SP3(4), SP4(4), SP5(4), SP6(4),
   SP7(4), SP8(4), SP9(4), SP10(4), SP11(4), SP12(4), SP13(4), SP14(4),
   SP15(4), SP16(4), SP17(4), SP18(4), SP19(4), SP20(4), SP21(4), SP22(4), SP23(4), SP24(4), SP25(4), SP26(4), SP27(4), SP28(4), SP29(4), SP30(4),
COMMON /REAC/ RP1(29), RP2(29), RP3(29), RP4(29), RP5(29), RP6(29),
   RP7(29), RP8(29), RP9(29), RP10(29), RP11(29), RP12(29), RP13(29),
COMMON /MIXT/ GP1(124), GP2(124), GP3(124), GP4(124), GP5(124), GP6(124)
COMMON /NOIZZ/ ZP1(64), ZP2(64), ZP3(64), ZP4(64), ZP5(64), ZP6(64),
   ZP7(64), ZP8(64), ZP9(64), ZP10(64), ZP11(64), ZP12(64), ZP13(64),
   ZP14(64), ZP15(64), ZP16(64), ZP17(64), ZP18(64), ZP19(64), ZP20(64)
COMMON /AREA/ ATPI(11, 2), PARAM(3, 12, 2), RTHCM(2), NSECT(2),
   NSECDU(2), ISHAPE(12, 2), RPFLS, NBL
COMMON /MSSFL/ SMAS, CMXXI, TSJ, PJ, IS(20)
COMMON /BL/ DLEBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2)
   SW, RO, JDIM, IPCINT
COMMON /MODPAR/ XMP1, DXM, FSTAG, CATFAC, TMODEL, XMODEP1, DXMOPD,
   TPLATE, XDIM
COMMON /MIXLIST/ IGA5, IGA6, NOIZZ, ICHANO
COMMON /MODPT/ TS1AM(20), TSAR(20), NT, MBL
COMMON /TAPOUT/ XXX(10), IOUT, NROUT, IFLW, ITPY, IMP, DATAPA
COMMON /BLOUT/ REPF, THTA(2), SN(2), XSN(2), PRREF, HR, QWDOT(2)
   TAUW(2)
COMMON /NEWMP/ FACMP, NMMDPDT
COMMON /RDWEDG/ ANG(10), RADDL(5), WX1, DWX, WX(20), TWEDGE, WK
   NW, NANGLE, NRADL, WEGDM, AXI SYM, ISWB
COMMON /AVG/ WSA
COMMON /TNE/ TN1(186), TN2(186)
COMMON /RDMOD/ LEWIS, IAMBIP
COMMON /READAT/ PRINS, DELFL, IPRNT, DELTIX, CXMAXI
   HSTAG, READXS, READG, AAMS, AXIMOD
   ICASE, NEELS, ISWB, INT, ICHAN, NQSI, NQO, NRECO, JCS(10)
++

NN

=a

aa

SZTFS!

I I I I I I

aaacraz a a ~ ~ o o o o o ~ ~ ~ ~ ~ z z z z z z z z z z z z z z z z
r r w k u u ~ ~ ~ ~ O O ~ ~ ~ ~ W W W L ~ ~ ~ O ~ ~ ~ ~ O O G O O O O O U U U U U U
aqaaaa ~ a a a a i ~ u a a ~ x ~ x r v u ~ w v v ~ ~ ~ ~ u ~ ~ ~ u u
aaaaq
WWWWW~JWWWWWWWWW
-.
urrua~aaaaaaaaaa

*

r

l

1c

l

OI
0
rn

0

13

CO
1c

r

N
Q)

r

•

0-1

(U

a@*

•
HN
e00
4 r r

c
.
(

*Od

l

FIN.

d

r

l

-\

l rr@

0

)L

.dN

a*.
- d o

*

U

Ir)

Wd(O

WO

dd

V)b
r

ChM

Z d

rc-

r r r

rr)

r
l

*9
0

r

rc
l

9
r
r(

d w d

l

rc r d
0 - 0

md

0

Wd

l( r

err

..
r

J.
•

(rCllN

*

0 0

l

-9

(COO

*

l

a9

...(
0 . 0

eb9

e r n aV)

.
.\ .

o

V

.-

dd

0.
0 .
-0

H.0

0 0

@ODIC
lr U
or,.
0

*

Q
.dD
VIF
Wo

0
b

.06+

H

d-

(CFl

•

..(-

*

N

d

*ma
obC *Ol
* r r *

03 0 -300
-PO

2

0.-

-

r r r

V)

r

**r~
0 rd
o m .

r

d9V)

rN9

l

0

* W e -

4.0

d U N

H...
-

\.

l

o

O H -

90
~ 4 - a

O H .

0
N

*'od

..

@-I*

l

0rfQ

l

0
Q

L

H

l

d

r,

l )Cmr

.*
l

CU

NNNNNNN--NNmmbbmM99bbm

~ ~ ~ ~ ~ ~ x ~ u ~ a ~ u a ~ u a u u ~ ~ : ~ ~ o o o o o o o o
~ ~ ~ ~ ~ ~ ~ ~ ~ ~ - ~ ~ - ~ u a a a a a a u ~ a a u

dc4Hw*dd
WNNN

*

d l- 0

+

- 0 6 ~ U l N

OM *

+

mb

r

•
r

l

N

r

•

0 - - 8 - *

* *

.*

..

+

*

r,

006
orre0
*OHl
it..
pC) b 0 9
Yl()d(De.
l
l
-27 l @11)69
lo*CD*
l 0
e * r I Ct -d
*NO.-Q-LDF
0
O N *
ad..d.-.
u e r . * N
0rr-0 0 c . d WrC - b - w
CU\
@ ( D o CUb*rfU...
O - r r r r .
-0-1
0
• d
H 0-t-n
-0.
o*Kl O@.o *NO+
0 e b .V)b
s + h
Mr. l
017U)rr
- 0
O C *\D - 0
IC)m l
* O H
- 9 O N0 - - 0 * - - - - \ * l•
s b .\N * d * - lo c r C d . - a
. d a d 0
0rC)D-O-eddlr)CUU
or).ON
00. l
* * nu O Nl- N Q F O m * ' a U l b ( U * b r r
a m *N
In - 0 - r r . C 0 - r
.NH.
r o d *In c r ( V r - c l r
d * d r ( d o
rm
Ha .dF-0.0
lO . \ V )
@(U . * . O d d
e m a n
oS--INcrm-we
* Q l l ) O * s 0-1
m .e-ou - 0 - ( U O ~.[11 - 4 . m ( c ~ a ~ )
l0 cO).

l

0
r

l

cc

*

Ul

r,

l

9

IC)

l

l

l

0

+

o

(y

m

d

C(

l

d

•

•

l

0
(C

+

(U

rc

d

.

ddd

l

•

o

N

r

**.

N

l

m~)rrl[)
a +a*
• dnc.
m

fQ

N

'in0

.*

l

rc

9

0 r+ d o
NnCU

0

O

l
4

We.
or.

r

•

cr
o
LO

0.4

c

Fl

l

-..lo
ccrN

l

d

*o
Npc-

H

r

OrtN

o

•

l

r o d N
00- r
MU l •
*NO
0

**

.-

0

l

l

C

r

c

a

•

Ul

rC

r
oa
d
r *
-dQ)rr
• • ON>- 0 4
• • 0-0
•
o
.
.
r
d
V
)
d
o
r
.
.
d
d
w
*Ie
Q
o
d
d
*
*
r
rc
u
0
0
l
r
l
d .rr
N N
wb.
*In lou om e m lc l.a- * m e 0 - .
,
r
cl
d.
N
r
r
l
r
N
OF@
~ r ~ ) r ~ g
0- a ~
--CI(UOIY
*cc r e c a l (V
w l 00
'b-3! 0" O d N 9 b . W d .crV)Ocr * * d
H
rr
-6
0 4 n m Xrr-9+
H orJdP3 m r r r
C r 0.03
* - * \ d o r * L d O Nlr r .c.Q 611) - 0 o d ~ rd H 0.d 6 l
I O b l a * - rcr
t
d 0
.OHe r ~ r r a 3 d d r .0SrC) - ( U - d F ) c - ~ c
l e
rr ON r. d
r(Cnll)m
or.0 9 0
N + . o ~ Q c , - m e ..dw
r r - . o * d ~ .
03
rctl
.nc....9
Olc-II).
(VbaiN
.m9mJ**--O(r--Crr.-~br...N
l
0 *S-NOdN
2
-9
-01) * S d d e - C d d
o ~ l n b
nph 0
~ l9
srl r
l0 - 0 0
0
r l
*
-Plcc * Z d l0 - 3
r r r Z lr r d b
r -dZP7.-r .P)rcr
0
00
* r r r
~ o a a - c n- 4 I
lL 0 - 0
O
O . ~U ~ - W H
ae- O H - .om
rrc* -0
r + r~ l
a+-CU l * +If)-*
.Ub+ OCUJM l e l U
- 6 . l.mLL
*- l
*d l
8 0 0 oO\O\C?
b O q r a - 0 0 wee l* l n E - e b * W - m u 7 0-o(r'90
d 000900
l 0 o O 4 U b Q oQ OUJ
*d
d.
*FIN 3 rN.4 * a * d Q d I - ( U *
a .a-Lnb-_) . U * N \ D * G J * O
a000 * 0 . O Q O O C .
rc r W 0 r N r r
rr
l d
l
OLL: 0 - 0
r r r o W b - 9 l0 - 0 *CO\@+OI o M r r l l ) + l n + O
l. a r c
0rc 0..
J o ~ n u r .r ~ - c z l ~ ~ c r o a -@
d r. r - - e r ) ~ - ~ . r - O ~ O F C O C U~ ( U V N O L O
-03-11) *r l LL'b
*+09 e *O
lO Q C M o
ll
*O r
o OHo b 0 u c . r *(Um(UdNrrd
r + t r *.-I
0 OQ-DLL *30ul~h,d@b@
* or- . l m c ) F o n I r . oO\m o r b - + O M
r
l
l
-044- O W N O c
0dNUl l m r d * r +
rrrL>i)90
l*LDrrGm
0 - 0
l- C ) ~ ' ? M 0
L ~0 0 0
lrCt l
r
5 0 c r N J l c O J(V r O - U e p - 0 r - 9 W V ) 004 r r * e m - * 0 0o
l0 0 o O
nJ .l.i~rnr- r~
rC)
.HW(U~MNW
m a u l
O M - u b a - 0 e m ~ c - r lr r r - 0 0 a ) d * o r..r
*Na * 0 - l l* L O & .O
n - 0 s l mGu' * W ) - d
c
l* ! I I ( V + - '
r lr I C U - o I lI d I
I lI lI
r I - 4 o w om
* ~ S C Z O U ) ~ O ~ * P ~ - I C L. bOI COd 3 . . r X X b .CJCUU l I 0 - 0
• *@ -0 l
*
a
0
*
+
r
c
l
u
)
q
e
m
o
a
+ m *e
N .w 1c.1 .rm\%CIU--(Uxocuco
.5-(\r*
oaa e a - r r u ~ a
( I O ~ U * O r~c r u z .
~3
OH
_J
.mzc r
.mmzml.dr(
0 0 0 - . - c u ~ ~ + c I N -O H
a r 1 w a d o r . o k w d r d Z U d @ . Z l i J ~ - 0 0 0 * r r f ' ) m r c l r0gC) r d - o W r C ~ 0 U c l W*&l(\IW c W r U
X r X Q
r r m u 16 - 0 0 3 4 :
l0 3 1 l e + Z C ( V l
ll. - - X U
o r lo m 9 - U + N * N 0 0 r d rt- -0
4 o U 1 -rr O N- 3 1 . + 0 u X b d O m I ( U ( U U l t 1 O F - Q ( n b .+I l0 - d O N0
0 0 le .Fl
ll
F IQ
\ r , u ~ b.ON O O ~ ~ O ~ ~ Y C L V ) N ~ * ~ O
O -( IV~ ~~ VU ~~ I ~~ N, Q
~* U
~
(O ( ~* T ) ( U ~* c ( ~ 8
ed ~ r c )
*
\b
I
@- *DJ\*e,
*rd\
-0~0\ r
\
-N
\
a\+\
- \ a \ * \ o\ * \
QD
d r
r ~ r Q OHM
d
-4-2
O r r d r r - N
r--rO
N 0
C(
0
l
NMO-r-r.
0-VIN 0 - O J r C ) la d J J ll* # i L I O b0 I ll.Q:QIf)
la * *(U *PC)OO 0 0 0 9 * b O a 3
~ C ( U O N a- o w a t - o m m a a n O U C L - H P I ~ Q
.n o t n u ~ r 1 ~ a r c r c ~ 0a- n4 oa a r a .a + n e a + a
mcc u . a m ~ c n ~ ~ ~ ~ ~ ~ ~ ~ N ~ ~ ~ ~ ~ ~ N ~ ~ C ~ O ~ ( ~ ~ ~ ~ H ~
d
N 3
0
W
W
zq
za
a a a a a u a a
lawa
wa
a 14
zc
a t
at~
t
t
t
t
t
t
~
QCN
t- u t
A4
Ja
q e a a a a a a
a ua
-4.
04
04
L

-0

r o d d o
~b l- 0
*N..l*
~ r c

--

OM
VI

0 .

+N

*

0.5

b.4.
l0 -

*

r

.- ..
- -\.... -.

d

.

.

+

.J"

..*

+

-

. .

.\

-

.-

,

.\

zt

ora0rNfQ4M

an

dN0

zo

QO

-NFIZo-NR

O

nN~bl()SbaOdNmdM

O

d

O

d

-

O

O

H

C

r

-

O

O

r

O


<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>2<em>1</em>0<em>0.0</em>6.1<em>5.1</em>16<em>1</em>4.1<em>13</em>1<em>14</em>1<em>15</em>1<em>4</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>2<em>1</em>0<em>0.0</em>6.1<em>5.1</em>16<em>1</em>4.1<em>13</em>1<em>14</em>1<em>15</em>1<em>4</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>7<em>E14</em>1<em>5</em>15.0<em>50.0</em>1.1<em>1.1</em>2<em>1</em>7.1<em>2</em>0<em>0.2</em>1<em>3</em>1.0<em>0.0</em></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>7<em>E15</em>1<em>5.0</em>50.0<em>1.1</em>1.1<em>2</em>1<em>7.1</em>2<em>0</em>0.2<em>1</em>3<em>1.0</em>0.0*</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>7<em>E14</em>1<em>5.0</em>50.0<em>1.1</em>1.1<em>2</em>1<em>7.1</em>2<em>0</em>0.2<em>1</em>3<em>1.0</em>0.0*</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>1</td>
<td>2<em>0.0</em>1<em>0</em>0.0<em>10</em>0.0/</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>DATA</td>
<td>REAC</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>RP44</td>
<td>044</td>
<td></td>
</tr>
<tr>
<td>RP45</td>
<td>045</td>
<td></td>
</tr>
<tr>
<td>RP46</td>
<td>046</td>
<td></td>
</tr>
<tr>
<td>RP47</td>
<td>047</td>
<td></td>
</tr>
<tr>
<td>RP48</td>
<td>048</td>
<td></td>
</tr>
<tr>
<td>RP49</td>
<td>049</td>
<td></td>
</tr>
<tr>
<td>RP50</td>
<td>050</td>
<td></td>
</tr>
<tr>
<td>RP51</td>
<td>051</td>
<td></td>
</tr>
<tr>
<td>RP52</td>
<td>052</td>
<td></td>
</tr>
<tr>
<td>RP53</td>
<td>053</td>
<td></td>
</tr>
<tr>
<td>RP54</td>
<td>054</td>
<td></td>
</tr>
<tr>
<td>RP55</td>
<td>055</td>
<td></td>
</tr>
<tr>
<td>RP56</td>
<td>056</td>
<td></td>
</tr>
<tr>
<td>RP57</td>
<td>057</td>
<td></td>
</tr>
<tr>
<td>RP58</td>
<td>058</td>
<td></td>
</tr>
<tr>
<td>RP59</td>
<td>059</td>
<td></td>
</tr>
<tr>
<td>RP60</td>
<td>060</td>
<td></td>
</tr>
<tr>
<td>RP61</td>
<td>061</td>
<td></td>
</tr>
<tr>
<td>RP62</td>
<td>062</td>
<td></td>
</tr>
<tr>
<td>RP63</td>
<td>063</td>
<td></td>
</tr>
<tr>
<td>RP64</td>
<td>064</td>
<td></td>
</tr>
<tr>
<td>RP65</td>
<td>065</td>
<td></td>
</tr>
<tr>
<td>RP66</td>
<td>066</td>
<td></td>
</tr>
<tr>
<td>RP67</td>
<td>067</td>
<td></td>
</tr>
<tr>
<td>RP68</td>
<td>068</td>
<td></td>
</tr>
<tr>
<td>RP69</td>
<td>069</td>
<td></td>
</tr>
<tr>
<td>RP70</td>
<td>070</td>
<td></td>
</tr>
<tr>
<td></td>
<td>18600</td>
<td>20000</td>
</tr>
<tr>
<td>---</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>C</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>D</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>G</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**DATA TN2**

<table>
<thead>
<tr>
<th></th>
<th>4.2</th>
<th>1.4</th>
<th>1.2</th>
<th>2.1</th>
<th>1.5</th>
<th>1.2</th>
<th>1.5</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1</td>
<td>1.3</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**DATA TN1**

<table>
<thead>
<tr>
<th></th>
<th>4.2</th>
<th>1.4</th>
<th>1.2</th>
<th>2.1</th>
<th>1.5</th>
<th>1.2</th>
<th>1.5</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1</td>
<td>1.3</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**DATA TN1**

<table>
<thead>
<tr>
<th></th>
<th>4.2</th>
<th>1.4</th>
<th>1.2</th>
<th>2.1</th>
<th>1.5</th>
<th>1.2</th>
<th>1.5</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1</td>
<td>1.3</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**DATA TN1**

<table>
<thead>
<tr>
<th></th>
<th>4.2</th>
<th>1.4</th>
<th>1.2</th>
<th>2.1</th>
<th>1.5</th>
<th>1.2</th>
<th>1.5</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1</td>
<td>1.3</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**DATA TN1**

<table>
<thead>
<tr>
<th></th>
<th>4.2</th>
<th>1.4</th>
<th>1.2</th>
<th>2.1</th>
<th>1.5</th>
<th>1.2</th>
<th>1.5</th>
<th>1.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.1</td>
<td>1.3</td>
<td>1.2</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Channel</td>
<td>Data</td>
<td>Notes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4H DCA/</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DCA</td>
<td>T22 CHANNEL, PROFILE 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1.1, 2.1, 2.1, 1.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1</td>
<td>NOZZ 12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-6.771664, 0.0, 2.646396, 0.0, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.0, 0.0, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0768428, -0.495987, 0.0, 17.779999, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>17.779999, 0.0, 15.240000, 2.304097, 0.176327</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0, 0.0, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.0, 0.0, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.0, 0.0, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.0, 0.0, 0.0</td>
<td>NOZZ 13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T12 CHANNEL FOR DCA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA CP1</td>
<td>/11.1, 12.1, 13HT12.1, 1.3HDC/</td>
<td>CHAN 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T22 CHANNEL FOR 10MW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA CP2</td>
<td>/11.1, 13.1, 13HT22.1, 1.4H10MW/</td>
<td>CHAN 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- NOZZ 12
- NOZZ 13
- CHAN 1
- CHAN 2
I

I

0
0

&

T L 2 0 / 9 5 * 3 r r 948. r 679.9
1000. r l a . Oa /
TL21/
1le3417E 00rla3311E OCrle32CbE 09rle3104E GOrle3006E
2 1 r 2 9 2 2 E 0 3 r l a 2 7 3 5 E 20 r l o 2 0 5 5 E 00rle25'8E
CO.lr2504E

Olrle7490E Olrle6253E
0 1 r l a 2 4 ~ 0 E0 1 r l a 2 5 5 D E
01rlrC070E 01r9e7600E
0092e75HGE Clr203950E
31rlo8C3CE Olrle7459E
Clrle44COE Clrla3930E
01rle138CE Olrle1670E
01 r 2 e 5 9 1 0 E 0 1 r 2 e 5 0 2 0 E
0 1 r 2 e 1 4 3 0 E 01 r 2 r O 6 7 0 E
01 r l o 7 2 5 0 E 0 1 r l s 6 B 3 0 E

Olrlo5320E
OlrleIR9OE
03e9e4800E
01r2a1949E
01rle6440E
Olrle342CE
Olr331090E
01 r2e42GOE
Olr l e 9 7 9 0 E
91 r l o 6 3 9 0 E

D A T A TL24/
1 89*1a
r3e60COE-01r3r00COE~dIr30OOOOE2 0 ~ 0 3 0 3 E - O l r e a O C G C E - C 1 s I ~ O C O C E0 0 * 1 e 2 0 C Q E 30.le403CE
OPr2e4000E OOrZePCCOE COr302000E 00r3e60COE
'ZeOZOCE
d 4 > 4 0 0 3 E '39 r 4 a 90COFrl 0 0 r 5 e 2 0 C O E 0 0 r S a 6 0 0 0 E 0 0 1
D A T A TC25/
1 70*2eOr
5e7000E
25s92C3E COrSo96COE 00r509900E 0 0 r 6 e O l C O E 00r6eC200E
3 6 e C 4 2 C E C 3 r 6 e 0 5 C O E 0 0 r 6 0 3 6 0 C E 0 0 r 6 e 0 7 0 0 E 00q6.OR30E
45e1202E 00r6el400E OCr6elhOOE 03r6a1903E 30rGe2300E

B 40*le/

01 .le9293E
31rle3420E
31rle046CF
03r80700CE
0lrle971CC
Olrle5100E
31rle23COE
01 r2e 6 7 3 E
Ol r 2 e 2 3 5 3 E
01 rla780GE

TL23/

12eZSbCE
21e394;E
3loC95CE
49.9QCOE
51e9540E
61e573CE
71e2523E
e2.777CE
42.289CE
SleY36CE

34T4

517

1 7 0 a E C3r l e 1 6 4 2 E 3 0 * 1 e 1 5 8 l E 0 0 ~ 1 1
e5 2 4 E 0 0 . 1 ~ 1 4 7 2 E
51e1376E 03rle1333E 00rlal292E 00rle1254E 00tle1219E
71 0 1 1 5 2 E Cbr 1 l l C Q 3 F 0 0 r l e l 0 3 9 E 09.1 e 0 9 9 3 E 00. 1 0 0 9 4 6 F
9 1 ~ 3 2 5 F EO C r l e ? E 3 4 E O C r l e O @ G 2 E 0 0 r l e Q 7 7 3 E 0 0 r l a 0 7 4 6 E
9 1 e C 6 3 7 E 3 0 r l e C b e S E C C * -67. e 6 7 r 2 8 * 1 e 3 r
e
2 0 2 1 1 0 E 91 r l o 9 2 6 9 E C l r l o 6 3 1 0 E
$1 e 4 1 0 Q E G l r l a 3 3 f . O E 0 1 rle27C-GE 01r1.2260E
01rla1820E
S l e O @ l O E O1rlrn?OO'
01r9a85GOt 03r9e4909E 00r9rl4COE
S S s 6 0 0 3 E 03r8e 38GOE 0 0 *Be 1600E 0 0 r7e9500E 0 0 1
D4TA TL22/
1 3 e 0 3 6 0 E O l r 2 e 5 9 9 0 E Cl r 2 e 3 3 3 C E O l r 2 e 1 5 2 0 E O l r 2 0 0 1 7 0 E
2 1 e 6 2 2 0 E 01rla74f30'
Clr19682CE Clrlo6243E Olrlo5290E
3 1 e 3 7 2 C E 3 1 r l e 3 2 9 0 E 0 1 r I e 2 P 3 0 E 0 1 r l e 2 4 0 C E 01 r l e 2 0 4 0 E
4 l a l 4 2 C E 0 1 r l m 1 1 3 0 t 01 r Z o 6 G S O E 0 1 r 2 e 0 4 0 3 E Cl r:e-r61GE
5 1 r 4 9 1 0 E 3 1 1 1 0 4 1 1 3 E 0 1 r l o 3 4 5 0 E Cl r l a 2 9 1 3 E C l r l e 2 4 4 0 E
61.1340E
0 1 r l e 3 7 9 0 E GI r l e G 3 2 0 E 0 1 r 9 * 9 5 S ' ? E C Q r 9 e 5 8 0 3 E
7 3 e 0 1 C O E 0 0 3 r E ! e 7 8 C O E O O * e * S 5 C C E 0 9 r R o 3 3 9 3 E 00.30533CE
8 2 r 3 5 C O E 0 1 r 2 r 6 2 3 0 E 01.204520E
01 r2e323CE C l r 2 0 2 1 C O E
o2eC3.iCE O l r I e G 7 3 ? E Cl'rle870C.E C l e l e 7 9 2 3 E O l r l a 7 2 4 Q E
01rle42ROE
s l s 6 3 0 C Z 3 1 r 1 * 5 5 1 C E 0 1 r l e 5 3 5 C E Ol.le4673E
S's333GE
01 r2ee240E Clr24513CF Glr202170E 0 1 r 2 ~ 1 6 7 C E
S l e 9 5 7 3 E 0 1 r l e 8 7 6 0 f 01 r l e 8 C 5 C E Cl r l e 7 4 3 0 E O I r l a 6 3 8 C E
s l e 4 S b G E 01.1e431CE Olrle3790F 0 1 r l e 3 3 6 3 E C I r l a 2 9 7 C E
Q l r 2 3 1 G E 0 1 r l e 1 9 9 0 E ClrZe232CE 0 1 r 3 e 0 7 9 0 E 01 r2e957CE
S 2 m 7 5 4 0 E 01 r ? o 6 b 5 0 E Clr2.566GE
91 r 2 e 5 1 0 i E C l r 2 e 4 2 7 C E
b 2 e 2 2 7 3 t 01r2e12'43E
GI r 2 e 0 3 3 C E 0 1 * 1 e 9 5 9 0 6 0 1 r l e 0 8 6 3 E
si..77,-~~
0: . I . ~ ~ G Q E 0 1 . I . ~ H ~ C E O I r ~ r 6 n 4~ /~ ~ ~

DATA
DATA


DATA TL35/100*1/  BD2 551
DATA TL36/  BD2 552
11.1960E00 1.2451E00 1.2906E00 1.2966E00 1.2665E00 1.2665E00  BD2 553
21.2455E00 1.2253E00 1.207E00 1.1919E00 1.1678E00 1.1496E00  BD2 554
31.1366E00 1.127E00 1.1197E00 1.108E00 1.1016E00 1.098E00  BD2 555
41.0958E00 1.0935E00 1.0925E00 1.0922E00 1.0922E00 1.0932E00  BD2 556
51.0923E00 1.0927E00 1.093E00 1.0933E00 1.0937E00 1.0937E00  BD2 557
61.0943E00 1.0944E00 1.0944E00 1.0943E00 1.0941E00 1.0947E00  BD2 558
71.0957E00 0.63*1/  BD2 559
DATA TL40/ 95*0.0* 1.2133* 1.65* 1.165* 0.0/  BD2 560
END  BD2 561
SUBROUTINE AESOLN (X)

LOGICAL ERR

ROUTINE FOR CALCULATING EFFECTIVE AREA RATIO AND ITS DERIVATIVE

CALL TO AESOLN (X) MUST BE PRECEDED BY CALL GEOMAR (X, S1, S2)

X = AXIAL COORDINATE ALONG NOZZLE

DELBL = BOUNDARY LAYER DISPLACEMENT THICKNESS DIVIDED BY ROAES

S1 = GEOMETRIC AREA RATIO (PROVIDED BY CALLING ROUTINE)

S2 = DERIVATIVE OF S1 (PROVIDED BY THE CALLING ROUTINE)

S2 IS RESET TO THE DERIVATIVE OF THE EFFECTIVE AREA RATIO

COMMON /RL/, DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SW*RD, JDIIM, JPCINT

DIMENSION DELP(2), DELPL(2)

COMMON /AREA/, ATP1(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12,2), NFRPL(2), JPRFLS, NBL

COMMON /AEgeom/, SORTA, S1, S2

COMMON /ERROR/ ERR

COMMON /NEOM/ OMDST(2), DDDELBL(2)

IF (NFRPL*EQ.2) GO TO 20

SORTAE = (SORTA - DELPL(1)) / OMDST(1)

S1 = SORTAE**2

S2 = (S2/SORTA - 2.*DDDELBL(1)) / OMDST(1) * SORTAE

GO TO 40

S1 = (S1 - DELPL(1)) / OMDST(1)

S2 = (S2 - DDELPL(1)) / OMDST(1)

GO TO 40

CALL GMAR3 (X, DYDX, DZDX, Y, Z)

IF (ERR) RETURN

D0 30 K = 1, 2

DELPL(I) = RO*DELBL(K)

30 DDELSP(K) = RO*(1. - OMDST(K))

DENOM = (RTHCM(1) - DDELSP(1)) * (RTHCM(2) - DDELSP(2))

FAC1 = DELP(1)

FAC2 = DDEL(2)

S1 = FAC1*(DYDX - DDELBL(1)*RO) + FAC1*(DZDX - DDELBL(2)*RO) / DENOM

RETURN

END
SUBROUTINE AGSOLN (AE, DEL, UPDOWN, AG, X)
LOGICAL ERR, FIX
SOLUTION FOR THE GEOMETRIC AREA RATIO WHEN THE EFFECTIVE AREA
RATIO AND BOUNDARY LAYER DISPLACEMENT THICKNESS ARE GIVEN
AE = EFFECTIVE AREA RATIO
DEL = DISPLACEMENT THICKNESS
UPDOWN = 1. IF DOWNSTREAM SOLUTION IS DESIRED
-1. IF UPSTREAM SOLUTION IS DESIRED
AG = COMPUTED GEOMETRIC AREA RATIO
COMMON /AREA/ ATP(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12,2), NPRFL(2), NPRLC, NBL
COMMON /NEO/ CMOST(2), DELPL(2)
COMMON /ERROR/ ERR
COMMON /BL/ DELPL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SW, RO, JDIM, IPCINT
DIMENSION DEL(2), DELPL(2), DSTP(2)
DATA RNAME, QHAGSOLN/
IF (NPRFLS.EQ.2) GO TO 20
IF (JDIM.NEQ.0) GO TO 10
C TWO-DIMENSIONAL FLOW
AG=DEL(1)+CMOST(1)*AE
GO TO 100
C AXISYMMETRIC FLOW
10 SORTA=DEL(1)+CMOST(1)*SORT(AE)
AG=SORTA+2
GO TO 100
C CHANNEL
20 DO 30 L=1,2
DSTP(L)=RO*(1.-CMOST(L))
30 DELP(L)=RO*DEL(L)
FIX=.FALSE.
40 YEO=RTHCM(1)*RTHCM(2)
AEC=(1.-DSTP(1)/RTHCM(1))*(1.-DSTP(2)/RTHCM(2))*AE-DELP(1)*DELP(2)
Y/20
A=AE
IF (COUNT=0) GO TO 50
50 CALL FINDX (A, UPDOWN, X)
IF (ERR) RETURN
IF (COUNT+1) GO TO 90
IF (COUNT+GT.20) GO TO 90
CALL GMAR2 (X, Y, Z)
IF (ICOUNT+GT.1) GO TO 70
A=AEC+Y*DELP(2)+Z*DELP(1)/YEO
X=X
PD=A-AE
IF (A*GE.1.) GO TO 50
20 X=0.
GO TO 60
70 CALL GEOMAR (X, A, DA)
IF (ERR) RETURN
F=AEC+Y*DELP(2)+Z*DELP(1)/YEO-A
IF (ABS(F)/A*LE.1DE-5) GO TO 120
IF (F*EQ*FO) GO TO 120
XL=X
X=X-F*(X-X0)/(F-FO)
XO=XL
FC=F
IF (X*UPDOWN*GE.0.) GO TO 60
C SIGN OF X IS INCONSISTENT WITH UPDOWN
IF (FIX) GO TO 90
IF (AE*GT*1.001) GO TO 90
C TROUBLE IS DIAGNOSED AS DUE TO SONIC POINT TOO FAR DOWNSTREAM
C OF GEOMETRIC THROAT. TRY FIX -- RESET DISPLACEMENT THICKNESSES
C TO THEIR VALUES AT THE THROAT.
DO 80 L=1,2
DELP(L)=DSTP(L)
FIX=TRUE.
WRITE (6,140)
GO TO 40
WRITE (6,150) AE,DELP,UPDOWN,X
NAMELIST /AGDMP/ OMST,RO,DSTP,DELP,Y0Z0,RTHCM,AEC,A E,AG,X0,F,FO,
Y2,Z1,COUNT
WRITE (6,AGDMP)
CALL DUMP (RNAME)
GO TO 130
IF (AG*GT*1.) GO TO 110
AG=1.
X=0.
GO TO 130
CALL FINDX (AG,UPDOWN,X)
GO TO 130
120 AG=A
130 RETURN
C
C IF (AE*GT*1.) GO TO 110
AG=1.
X=0.
GO TO 130
CALL FINDX (AG,UPDOWN,X)
GO TO 130
120 AG=A
130 RETURN
C
C FORMAT (29H0+++++ FIX REQUIRED IN AGSOLN)
C FORMAT (30H0CONVERGENCE FAILURE IN AGSOLN,9X,3HAE=,1PE12.5,8X,4HEAGS
1L=2E12.5,8X,7UPDOWN=,0PF4.1,8X,2HX=,1PE12.5)
END
SUBROUTINE AXFIT

REAL ACOM(30), ELMNT(10), HP(20)

DETERMINES AXIAL POSITION DURING PERTURBATION SOLUTION

DOUBLE PRECISION AA, CAPX, CDI

COMMON AA(22,24), CDI(20*10), CAPX(20)

1 C, CARB, CH, CHA, CL, CM, CMA, CRA, CRP

2 CCR, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMX

3 CTP, CTPL, CTT, CX, CXT, CMAX, DATEST, DBTEST, DELT

4 DELT2, DELTAX, DLOGR, DT, ENT, FLUX, HDELX, PCT

5 SPCTEST, PRFSA, PRFSD, PRESTH, PRHO, RHAP, RHQ, RHOB

6 SHQBAR, RHQ, RHOP, RHPL, RHMT, ROBAR, ROBAP, SCG, SDT

7 SL, SHPG, SC, SL64, SM, SU, SUMG

8 TEST, TESTB, TSTOP, UP, ZP, ZPA

COMMON EE(64), BET(20), BLBK(31), CAI(64), CAPXTH(20)

1 CCPJ(20), CEA(64), CGI(20), CMU(20), CHA(64), CHJ(20)

2 CLNK(64), CLNP(64), CMW(20), ETAI(64), ETAJ(20), GJA(20)

3 GJU(20), PERTGJ(20), PGJ(20), PI(64), PICHI(64), QM(20)

4 QO(64), SAJJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20)

5 SKU(20), SS(20), TH(30), TFA(20), TFD(20), TFC(20)

6 TFD(20), TFE(30), TKV(20), XMJAT(20), XNIU(64)

COMMON ETAI(64,20), ELJ(10,20), GELJ(10,20)

1 XNU(I(64,20), XNUIP(64,20)

COMMON IC, IM, INEQ, INEQV, IP, IRUN, ISC, ISCPI

1 ISMC, ISVCMR, ISR, ISW, NP, ISNP, ISP2, ISP3, ISP4

2 ISW1, ISW2, ISW3, ISW4, ISW5, ISW6, ISW7

3 ISW8, ISW9, ISW10, IUPO, IZERO, JJK, LC, ML, NFIT

4 NI, NN, NNN, NNS, NOT, NTEST

COMMON IGJ(30), IGM(20), ITB(5), KUR(64,20), LPII(20,10)

COMMON ACOM, ELMNT, HP

COMMON /HL/ DELCL(2), BLINT(2), XZERO, TWALL, CPAWALL, VISROT, DIAM(2)

IF (CT-CTMAX) 20, 20, 10

UPDOWN=1

GO TO 30

UPDOWN=0

IF (ISW3B, NE, 0, AND, UPD, EQ, 0) GO TO 40

CALL FINDX (AFNTS, UPDW, C)

GO TO 39

CALL AGSOLN (AFNTS, DELBL, UPDOWN, AG, C)

RETURN

END
SUBROUTINE AXSSEC (LL, NN)
SUBROUTINE AXSSCD ADDS DATA FOR UNSPECIFIED SPECIES PAIRS TO STEP AXS
LL OF THE EDTEC CROSS SECTION DATA. SPECIES TO BE CONSIDERED ARE AXS
GIVEN IN NUMERIC ORDER IN THE FIRST NN ENTRIES OF THE ARRAY I. AXS
COMMON /TRAN/, 1, O3(20, 20), ZM2(20) AXS
COMMON /TRANS/ KKQ(100), NNQ(100), ISEQ(100), NNKQ, I(50) AXS
COMMON /TRANS7/ V(400), KQ(100), NQ(100), IQ(400), JG(400), NKQ AXS
IF (NNLE.0) GO TO 40 AXS
LQ1=NQ(LL)+1 AXS
N=0 AXS
DO 20 IV=1, NN AXS
II=I(IV) AXS
DO 20 JY=IV, NN AXS
JJ=I(JV) AXS
IF (Q11, JJ, NEO) GO TO 20 AXS
Q11, JJ = 1 AXS
N=N+1 AXS
LQ=400 AXS
DO 10 L=LQ1, 399 AXS
LQ=LQ-1 AXS
10 JQ(LQ+1)=JQ(LQ) AXS
JQ(LQ1)=II AXS
JQ(LQ1)=JJ AXS
CONTINUE AXS
DO 30 L=LL, NKQ AXS
30 NQ(L)=NQ(L)+N AXS
40 RETURN AXS
END AXS
COMMON /SS/ CAS.US
COMMON /SWITCH/ XPB,DXPB,W
EQUIVALENCE (XP,XPU),(DXP,DXPB)
COMMON /DRV/ SHJGJ
COMMON /TNCE/ SUMGH,SCPGH,QQPR,QQPE
COMMON /BLRA/ YQ70
COMMON /MCDP/ MODLP
COMMON /CCLOSP/ CGMW,OPJ(10),IJC5(10),NCS
COMMON /TNONEO/ CTE,DTE,EPAR,EPAR(2,25),NT,ITR(25),KTF(25)
1     KTR(25),ICHX(25),IPA(25)
DATA ONE /1,0/, ZERO /0/
1     HREF,HW,TE,TREF,CCSB,DAQ,DYDX,SY,OMEGA,SIGN,DM,OM,US
2     PR,PRD,PR,OMEGA,ORDIN,X,Omega,DM,OM,US,RO
3     SW, NBL, L_TYPE, RFR, BLIGHT, ENRIM, SN, XSN, THETA1, THETA, HTR
4     DELHL, DELBL, XL, RAFT, TAM, WDOT, XTEST, GAMMAH, WW, AVCON, B, EK1
5     OLMF, TRCALC, TEP, TREF, RMR, F0, F2
DELB=DISPLACEMENT THICKNESS
THETA=MOIEMENT THICKNESS
THETA1=MOIEMENT THICKNESS FOR TWO-DIMENSIONAL FLOW
THE BOUNDARY LAYER THICKNESSES DELHL, THETA, THETA1 ARE ALL
NODIMENSIONALIZED USING THE GEOMETRIC THROAT RADIUS RO THEY
ARE CONVERTED TO INCH UNITS FOR OUTPUT, IN SUBROUTINE OUT1.
IF (XGP*OR*, IPOINT) EQ 0) GO TO 10
IF (XEQ CXH) GO TO 550
WRITE (6,560) X, XP
WRITE (6,550)
GO TO 550
10   DO 540 L=1,NPRFLS
540 IF (IPOINT NE OR L) EQ 2) GO TO 90
     B=2.1411.28*SWP1+0.93*(SWP1**2)**2
     SM1=B-1.
550   EK1 AND B ARE THE COEFFICIENTS IN THE COHEN-RESHOTKO LINEAR FIT TO
      THE MOMENTUM PARAMETER (CAPA, N) AS A FUNCTION OF THE CORRELATION
      NUMBER SN (SMALL N). THE FORMULA IS N=EK1+B*SN.
      SYFAC=SWP1*(7.2+3.1*4SWP1)
      SYFAC2=1.455*SWP1**0.9
      HCON=CRP/CMA
      MW=CPWALL*TWALL
      TEO=0.
      TREFP=0.
      HO=CMY*HCON
      DO 20 I=1,ISS
20     XSAVE(i)=SAVEC(i)
      XTEST(i)=1.
      DO 40 I=1,ISS
40     DO 30 K=1,NCS
30     IF (IJC5(K) NE 1) GO TO 30
      SAVEC(I)=OPJ(K)
      GO TO 43
30     CONTINUE
      SAVEC(I)=0.
50     CONTINUE
      CALL TRANSP (TWALL,PRESA)
50  DO 50 I=1,ISS
   SAVEC(I)=XSAVE(I)
   IF (NPRL*EQ.1) GO TO 60
   NDZ0=THCM(1)*RTHCM(2)
   ITYPE=3
   GO TO 80

60  IF (JDIM*EQ.0) GO TO 70
   ITYPE=2
   GO TO 80

70  ITYPE IS A FLAG FOR TYPE OF NOZZLE GEOMETRY
   ITYPE=1  TWO-DIMENSIONAL
   ITYPE=2  AXISYMMETRIC
   ITYPE=3  RECTANGULAR CHANNEL

80  AVCON=1/R0
   R0 IS SET IN MAIN
   COUPLED=FALSE*
   ISMO=30
   DDEBL(I)=0.
   DDEBL(2)=60.
   90  IF (L*EQ.2) GO TO 100
   IF (NT*EQ.2) HC=CHA*HCON
   AS=SORT(CAS*CT)
   100  IF (FINAL) GO TO 110
   H.INTI=PLINT(L)
   XI=XI(L)
   110  IF (L*EQ.2) GO TO 220
   TE=CTAP*T
   DO 120 I=1,ISS
      IF (AM*(XTEST(I)-SAVEC(I))*GT.0.01) GO TO 130
   CONTINUE
   CLOM=TRUE.
   TRCALC=FALSE.**
   DO 150 I=1,ISS
   120  CONTINUE
   CLOM=FALSE.
   DO 140 I=1,ISS
   130  CONTINUE
   XTEST(I)=SAVEC(I)
   GO TO 160
   140  IF (ABS(TC-TEP)/TE L*0.005) GO TO 180
   CALL TRANSP (TE,P*PRESA)
   TEP=TE
   IF (EFR) RETURN
   IF (IPOINT*LE.1) PRRF=PR
   DO 170 I=1,4
   150  CONTINUE
   TRPSV(I)=TRP(I)
   160  IF (EFR) RETURN
   IF (IPOINT*LE.1) PRRF=PR
   DO 170 I=1,4
   170  TRPSV(I)=TRP(I)
   180  ME=HCON+H
   PPRW=PPRF**0.56
   HR=ME+PPRW*(H-HR)
   ME=RECOVERY ENTHALPY
   HREF=(HR+H0)+0.22*(HR-HR)
   TREF=CTAP*HREF/H0
   CALL GEMAP (X,AG,CADX)
IF (ERR) RETURN
AG=GEOMETRIC AREA RATIO (AREA DIVIDED BY AREA AT THROAT)
AE=EFFECTIVE AREA RATIO ALLOWING FOR BOUNDARY LAYER DISPLACEMENT
THICKNESS
GO TO (190,200,210) ITYPE

190
AJ=1
COSB(I)=1./SQR((1.+(RO*DADX)**2))
GO TO 230

200
AJ=AG
COSB(I)=1./SQR((1.+(RO*DADX)**2)/AG)
GO TO 230

210 CALL GMAR3(X,DYDX(1),DYDX(2),YZ(1),YZ(2))

220 LPR=LPRIME(L)
AJ=YZ(LPR)*2
COSB(L)=1./SQR((1.+(DYDX(L)))**2)

COSB=COSINE OF THE ANGLE BETWEEN THE NOZZLE WALL AND THE AXIS OR CENTER PLANE

230 IF (L.EQ.2) GO TO 270
UPRIM=U*US
IF (DLCMF*ABS(TREF-TRFP)/TREF*LT.0.005) GO TO 240
CALL TRANSP(TREF,P*PRESA)
IF (ERR) RETURN
TREF=P=REF
TRCALC=TRUE
PRREF=PR
VISC=VISC

240 IF (*.NOT.,TRCALC) GO TO 250
VR=VISC/VISCW
T=TRF/TWALL
OME=ALOC(VR)/ALOC(TR)
IF (IPOINT*NE.1) GO TO 250
EK1=0.38-0.76*PRREF*(1.-CMEG1)*EXP(-6.76*SWPI)

250 IF (EK1.GE.0.01) GO TO 250
WRITE (6,530)
WRITE (6,530)
E=0.01

260 DO 260 I=1,4

260 TRPI(I)=TRPSV(I)
AM=UP/ME/AS
ORDN(L)=AJ*AM**SM1*AS*RH*T
IF (ITYPE.EQ.2) R=SQR(A)
R=RATIO OF LOCAL NOZZLE RADIUS TO THROAT RADIUS

C FIRST POINT
DX=(X-XZERO)/10.
DX50=DX**2
XX=XZERO
X1(L)=0
PL1NT(L)=0
C=AM*AG
CALL RADIUS(ITYPE,XZERO,R1,AG1,AGJ,L)
AY=CY/AG1
ORD1=AGJ**SM1
DO 280 I=1,10
XX=XX+DX

280
CALL RADIUS (ITYPE, XX, R2, AG1, AGJ, L)
DXI=SQRT(DXSO*(R2-R1)**2)
XI(L)=XI(L)+DXI
AM1=CG/AG1
ORD2=AG*AM1**BM1
BLINT(L)=BLINT(L)+0.5*(ORD1+ORD2)*DXI
ORD1=ORD2
280 R1=R2
BLINT(L)=BLINT(L)*AS
DLIM(L)=DADX/AG
C X IS THE AXIAL COORDINATE DOWN THE NOZZLE, ZERO AT THE GEOMETRIC THROAT.
C XI IS THE STREAMWISE COORDINATE IN THE BOUNDARY LAYER, ZERO AT THE NOZZLE ENTRANCE (WHERE X=XZERO).*
GO TO 370
290 DX=X-XP
IF (INEQ.EQ.0) GO TO 320
C COMPUTE DERIVATIVES FROM RATE DATA
C IF (NT.EQ.1) GO TO 300
DUDX=GJ(DCPJ(J))DTE+SCPGH*DT
GO TO 310
300 DUDX=SCPG*DT
310 DUDX=DMA/UX(DUDX+SHJDG)
IF (NT.EQ.2) DUDX=DUDX+DCHA/SU
DLIM(L)=DUDX/U-0.5*DT/T
GO TO 360
320 IF (IPINT.GT.2) GO TO 340
330 DDX=(AM-AMP)/DX
GO TO 350
340 IF (DX.LE.0) GO TO 370
DX2=DX+DXP
XPXP=X+XP
DDMX=(AM-AMP)/DX
C2=(DDMX-(AM-AMPP)/DX2)/DXP
C1=DDMX-C2*XPXP
DDMX=C1.2*C2*X
IF (DDMX.LT.0) GO TO 330
350 DLIM(L)=DDMX/AM
C TRAPEZOIDAL RULE INTEGRATION
360 DXI=DX/COS(L)
XI(L)=XI(L)+DXI
BLINT(L)=BLINT(L)+0.5*(ORDIN(L)+ORDINP(L))*DXI
370 DLIM(L)=DLIM(L)*COSBL(L)
RDH=RH*RHAP*UPRIM*RO/VISC
C RER=REYNOLDS NUMBER BASED ON THROAT RADIUS AND FREE-STREAM CONDITIONS.
C ENPRIM=BLINT(L)/ORDIN(L)/RO*EK1
RM=RWM*W/VISC/RH
THETA1=SQRT(ENPRIM/RER*RM)
GO TO (390, 390, 380, I, ITYPE
380 THETA(L)=THETA1
GO TO 400
C TRANSVERSE CURVATURE CORRECTION FOR AXISYMMETRIC NOZZLES
390 SQ=SQRT(1.+2.*COSB(L)*THETA1/R)
THETA(L) = R/COSB(L)*(1-SOT)

IF (NOTFINAL AND IPOINT=NE) GO TO 430
SN(L) = ENPROM*0*DLM(L)
IF (IPOINT=NE) GO TO 410
XSN(L) = SN(L)
GO TO 420

410 WW = EXP(-AVCON*D/XAG)
XSN(L) = (1-WW)*SN(L)+WW*XSN(L)

420 XSNV = AMONI(XSN(L),ZERO)
ALPR = ALOG(PRREF)
GAMMAE = AS**2/(3.141597/CM*TE)
IF (GAMMAE < 1) GAMMAE = 1.01
SIGH = 1/((1+2*(GAMMAE-1.0)/AM**2))
HTR(L) = 0.75*(EXP(4.6*XSNV)*(1+3.0-15.6*XSNV)*WPI-1.0)
HTRP(L) = HTRP1(L)-1.0

430 HFP1 = HTRP1(L)/T
HFP = HFP1+1.0
DEBL(L) = HFP*THETA(L)
IF (IPOINT<LE=1.0 OR NOTFINAL) GO TO 470
IF (COUPLO) GO TO 440
IF (IUPDOEQ=1) GO TO 470
COUPLO = TRUE.

440 DDELBL(L) = DDEL(L) - DDELBL(L)/(X-XP)
IF (ISMOD=EQ=1.0 OR LE=EQ=2) GO TO 450
WD = 1/ISMD
GO TO 460

450 IF (L=EQ=2) GO TO 460
WD = 0.5*AMIN(DONE/(DX/DXO)**2)
460 DDELBL(L) = WD*DDELBL(L) + (1-WD)*DDBLP(L)
IF (ISMD=EQ=1.0 OR L=EQ=LEPRT) GO TO 470
ISMD = ISMD-1

470 CONTINUE
IF (FINAL) GO TO 480
BLINT(L) = BINTI
X1(L) = X1I
GO TO 540

480 DELBLP(L) = DDELBL(L)
JNP(L) = ORDIN(L)
OM = AMAXI(1.0-OMEGA,ZERO)
F0 = EXP(-0.427*(SWP1**(-OM)-1.0))
AX = 15.5*EXP(-29.0*SWP1)+0.67*EXP(4.37*SWP1)
BX = 0.7+0.47*SWP1
FI = 0.0*OM1**1.2*(1.0-SQRT(1.0-SIGN))
IF (SN(L)==LE=1.0-2) F1 = F1/(1.0+AX*(-SN(L))**S8X)
IF (FC) 490, 490, 500

490 F3 = 0
GO TO 510
500 F3 = EXP(3.5*AMIN1(SN(L),ZERO)/F0**1.57)
510 F2 = F0*0.45*OM1*(1.0-F3)*F1
IF (SN(L) == LE=1.0-2) GO TO 520
XL = C+2.265
GO TO 530
520 XL = C+2.265+SWFAC2/(2.7/SN(L)+5.06/(-SN(L))**0.224)
530 XL = F1+XL
GO TO 510
TAUW(L) = 0.0209 * XL * RMR * VISC * UPRIME / THETA(L) / R0
RA = SWP1**(-0.86 * OMEGA) * (1 + (6.2 + 2.35 / SWP1**0.86) + (1 - PRREF)**2 * (0.15 * SWP1**0.38 * SIGM / OMEGA))
1155*(1 - 2 * SWP1) + 0.29 * SWP1**0.38 * SIGM / OMEGA)
R3 = 0.94 - 1.31 * SWP1**0.486 - ALPR**((0.62 * SWP1**3.5 * DM1 - 0.49 - 3.61 * SWP1)**3)
1 * SIGM)
RAF = PRREF**(-0.35) * (1 - SN(L)) / (RA - RB * SN(L))
QDOT(L) = 1760 * TAUW(L) * (HR - HW) / UPRIME / PRW / RAF
DDBLP(L) = DDELBL(L)
IF (L(RN - NFRS)) GO TO 540
IPOINT = IPOINT + 1
AMP = AMP
AMP = AM
X = X
DX = DX
IF (* NOT * MODLPT) DX = DX
REPF = REP * R0 * 30 * 48
CONTINUE
BLA 350
IF (ISW4B > 0) WRITE (6, DMP)
IF (ISW4B < 0) WRITE (6, 570) X, IPOINT, ISMD, WD, (DDELBL(J)), J = 1, NFRS
540
RETURN
550
FORMAT (36H0 ZERO OR NEGATIVE STEP IN BLAYER, X = 1PE15.8, 10X, 3HX = IE15.9)
560
FORMAT (12H BLAYER X = F10.5, 5X, 7H IPOINT = 14, 5X, 5H ISMD = 12, 5X, 3H WBLA 350
10 = F6, 4, 5X, 7H CDELBL = 2F15.6)
570
FORMAT (56H0**** CALCULATED EK1 LESS THAN 0.01 EK1 RESET TO 0.01)
580
1)
END

ORIGINAL PAGE IS OF POOR QUALITY.
SUBROUTINE BLCCALL (FINAL)
CALLING ROUTINE FOR BOUNDARY LAYER IN NCNEQUILIBRIUM CALCULATIONS

REAL ACOM(30), ELEMENT(10), HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22, 24), CDIJ(20 + 10), CAPX(20)

COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO, BLB
C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP, BL
2 CRRH, CR, CST, CT, CTP, CTB, CTC, CMAX, CTMX, CT
3 CTP, CTPL, CT, CX, CXB, CMAX, DATEST, DBTEST, DELT1
4 DELT2, DELTAX, DLOGA, DLOGC, DT, DNT, DFLUX, HDELX, PCT
5 PCTest, PREG, PRESB, P prep, PREH, RMHAP, RH, RHO, RHCM
6 RMOBAR, RMOC, RHOP, RHPL, RHTM, ROBAR, ROBARP, SPG, SBT
7 SEN, SP, SG, SC, SL, SL64, SM, SU, SUMG
8 TEST, TESTB, PRINT, TSSTOP, UP, ZP, ZPA
9 COMM
10 BLC
11 IF (1NEQ, OE, 0) GO TO 20
BLC
12 IF (ISk3 YetQ.0) GO TO 50
BLC
13 IF (1hEO. NE, 0) GO TO 60
BLC
14 CT=CT+PCT
BLC
15 WHO=G4C-PRHO
BLC
16 TUR=TURN
BLC
17 END ecc

COMMONIC, IM, INEG, INEOV, IP, IRUN, ISC, ISCP1, BLC
IC, ISCP, ISMCN, ISR, ISS, ISSN, ISSP1, ISSP2, ISSP3, ISSP4, BLC
1 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, BLC
2 ISW5A, ISW5B, IUPO, IZERO, JJK, LC, M1, NFIT, BLC
3 NIT, NNN, NNS, NOS, NOT, NTB
COMM
10 IGI(20), 1GM(20), 1IT3F5, KUR(64, 20), LPIJ(20, 10), BLC
COMM
10 ACOM, ELEMENT, HP
COMM
10 DIMENSION AAA(22, 24), BTA(64, 20), CAPO(31), CCI(20), DGJ(20), GJ(20), SBPL
COMM
10 JJ(20), SDGJ(20), SHJAP(20), THEV(20), THEVP(20)
COMM
10 EQUIVALENCE (AA(1, 1), AA(1, 1), (BEtA(1, 1), BTA(1, 1)), BLC
COMM
10 (BLH(1), CAPQ(1), ) (CAPXTH(1), CCI(1), ) (GJA(1), DGJ(1), )
COMM
3 (CAPX(1), GJ(1), ) (SAJ(1), SDGJ(1), )
COMM
3 (SHJA(1), SHJAP(1), ) (THEV(1), THEVP(1))
COMM
10 /TEMPY/ SAVFC(20)
COMM
10 IF (INEQ, OE, 0) GO TO 20
COMM
10 ID10 =1, ISS
COMM
10 SAVFC(J) =GJ(J) *CM
COMM
10 GO TO 40
COMM
10 CT=CT+PCT
COMM
10 RHO=RHO+PRHO
COMM
10 DO 3C =J, 1, ISS
COMM
10 SAVFC(J) = (GJ(J) +PERTGJ(J)) *CM
COMM
10 IF (ISW3B. EQ. 0) GO TO 50
COMM
10 CALL BLAYER (FINAL)
COMM
10 IF (INEQ. NE, 0) GO TO 60
COMM
10 CT=CT-PCT
COMM
10 RHO=RHO-PRHO
COMM
10 RETURN
COMM
10 END
SUBROUTINE BXSECT (L,MV,I,J)

C DETERMINES THE CORRESPONDENCE BETWEEN THE KO AND BKX
C ARRAYS, AND SEARCHES THE BV ARRAY FOR REFERENCES TO SPECIES
C OR ION

V(400)*KQ(100)*NQ(100)*JO(400)

C COMMON /TRANS7/ NV(15)*N

L=0

RETURN

END

C

10 IF (L.EQ.0) GO TO 30

L=L+1

K=KQ(L)

IF (K.EQ.9) GO TO 30

IF (K.EQ.12) GO TO 50

MADD=0

20 MADD=MADD+2

30 MY=MV+MADD

I=V(MY+1)

J=V(MY+2)

RETURN

40 V(MY+1)=I

V(MY+2)=J

IF (MV.EQ.0) GO TO 20

50 MV=MV+2

IF (MV.NE.0) GO TO 10

RETURN

END
SUBROUTINE COMM

LOGICAL ERR, FAILED, LOGC, NORAC, RACLC

REAL ACOM(30), ELMENT(10), HP(20)

DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ

COMMON AA(22, 24), CDIJ(20, 16), CAPX(20)

COMMON A, APNOT, AFNX, AJACH, AR, ARBA, ARBB, BZERO, 

1, C, CARB, CH, CHAR, CMK, CMA, CHA, CRP, 

2, CRRB, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMX, 

3, CT, CTPL, CTY, CX, CXG, CXMAX, DATEST, DBTEST, DELT1, 

4, DELT2, DELTAX, DLOC, DLOW, HFLA, HFLUX, HPL, 

5, SPCTEST, PRES, PREA, PREB, PRDC, PWM, RMA, RPM, RH, 

6, RMOBAR, RHOC, RHOP, RPPL, RPFL, RTIG, RTDA, RTG, DT, 

7, SIN, SHPG, SC, SL, SLG4, SJ, SU, SU2, SUMG, 

8, TEST, TPR1, TPR2, TSTOP, UP, 

9, BE(64), BET(20), BLOC(31), CAI(64), CAPTXH(20), 

10, CCPJ(20), CFACF(64), CI(64), CMMU(20), CHI(64), CHII(20), 

11, 2CLNMC(64), CLNPI(64), CMW(20), ETA(64), ETAJ(20), 

12, GJ(20), PERTJG(20), PGJ(20), PI(64), PICH(64), QM(20), 

13, QPI(65), QAJ(20), SCJ(64), SEP(20), SEJ(20), SHJ(20), 

14, SKIL(20), SII(20), TII(30), TIIA(20), TFFB(20), 

15, TFS(20), TFE(20), TFIN(20), TF(20), THEV(20), XJMAT(20), 

16, XUUIJ(64, 20), XUUIJP(64, 20)

COMMON IC, IM, ING, INFGV, IP, IRUN, ISC, ISCP1, 

1, ISMC, IMLCPN, ISR, ISS, ISSN, ISSP1, ISSP2, ISSP3, ISSP4, 

2, ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A, 

3, ISW6A, ISW6B, IUPG, IRFG, IJK, LC, MI, NFIT, 

4, NIT, NNNS, NOS, NIT, NTEST, 

COMMON IGJ(20), IGM(20), ITHB(5), KUR(64, 20), LPIJ(20, 10), 

COMMON ACOM, ELMENT, HP

DIMENSION AAA(22, 24), BTA(64, 20), BAPQ(31), CCI(20), DGI(20), GI(20), SB, 

1, SJ(20), SDGJ(20), SHJ(20), SIS, SISN, SINS, SISN, SISN, SISN, SISN, 

1, (BLK(1), CAPO(1)), (CAPX(1)), (CCI(1)), (GJ(1)), 

2, (CAPX(1)), (CJ(1)), (CJ(1)), (CJ(1)), (CJ(1)), 

3, (SHJ(1)), (SHJ(1)), (THEV(1)), (THEV(1)), 

COMMON ERR/ERR, FAILED

COMMON /ERROR/ FAILED

COMMON /NONEG/ CT, DTE, EPAR, EPAR(2, 25), NT, ITR(25), KTF(25), 

1, KTR(25), LCH(20), IPA(25), 

COMMON /AREA/ ATP(11, 21), BYRAM(3, 12, 2), RTHCM(2), NSEC(2), 

1, NSEC(2), ISHAP(12, 2), NPROFL(2), NRPL, NBL, 

COMMON /TNC/ XMJATD(20, 20), TNLTP(2, 2), CTD(2), 

COMMON /TNC/ SUMG, SPCG, QDR, GDFE, 

COMMON /GLM/ NORAC, 

DATA V1C, V1C2 /1, 1.094426E+1, 13288E-16/, VNC /4, 99E29/

DIMENSION ALGJ(20)

EQUIVALENT (ALGJ(1), ALGJ(1)), (HETA(1)), (DGA(1)), 

1, (CAPX(1)), (CAPX(1)), (CAPX(1)), (CAPX(1)), 

2, (CAPX(1)), (CAPX(1)), (CAPX(1)), (CAPX(1)), 

3, (CAPX(1)), (CAPX(1)), (CAPX(1)), (CAPX(1)), 

COMMON /ERROR/ ERR

COMMON /ERROR/ FAILED

COMMON /NONEG/ CT, DTE, EPAR, EPAR(2, 25), NT, ITR(25), KTF(25), 

1, KTR(25), LCH(20), IPA(25), 

COMMON /AREA/ ATP(11, 21), BYRAM(3, 12, 2), RTHCM(2), NSEC(2), 

1, NSEC(2), ISHAP(12, 2), NPROFL(2), NRPL, NBL, 

COMMON /TNC/ XMJATD(20, 20), TNLTP(2, 2), CTD(2), 

COMMON /TNC/ SUMG, SPCG, QDR, GDFE, 

COMMON /GLM/ NORAC, 

DATA V1C, V1C2 /1, 1.094426E+1, 13288E-16/, VNC /4, 99E29/

DIMENSION ALGJ(20)

EQUIVALENT (ALGJ(1), ALGJ(1)), (HETA(1)), 

1, NAMLIST /CC/ DNP/ CAPX, RHO, DLOG, SUMG, SFIN, RPM, CRP, 

2, SU2, SFIN, SFIN, SFIN, SFIN, SFIN, 

3, QDR, QDFE, KFR, RKF, ELAS, CDFN, CDFP, 

4, DATA ICG/0/, ONE /1/, 

DATA ICG/0/, ONE /1/, 

ICOUNT = 1COUN+1

IF (ISW54.LT.0) ICYCLE = -ISW58
COMPUTATION OF Pichi

LOGC=F*TRUE*
IF (NTi*EQ*1) GO TO 160
KF=KF*I(I)
KR=KTR(I)
GO TO 170
KF=1
KR=1
170 IF (KF-3) 180,190,200
180 CLKF=ETA(I)*CLNTD(KF)+CAI(I)-CEACT(I)/(CTD(KF)*CRP)
GO TO 240
190 CLKF=ETA(I)*CLNTD(2)+CAI(I)+ALOG(1-EXP(-CEACT(I)/(CTD(I)*CRP)))
GO TO 240
200 IF (R CALC) GO TO 220
R CALC=**TRUE*
IF (NPRFLS*EQ*2) GO TO 210
CALL GMAR (CX,R)
GO TO 220
210 CALL GMAR2 (CZ,Y,Z)
R=1./SORT(0.5*(1./Y**2+1.+Z**2))
220 IF (KF*EQ*4) GO TO 230
CLKF=CAI(I)-0.5*ALOG(R)
GO TO 240
230 IPA=IPA(I)
TAU=SAPR*RHDP*GJ(IPAI)*R
CLKF=ETA(I)*CLNTD(2)-ALOG(AMAX1(ONE,TAU))+CAI(I)
240 CLNP1(I)=CLKF*Z1
CLKR=CLKF
CLNM1C(I)=0.0
T6=0.0
T7=0.0
IF (KR*EQ*0) GO TO 250
Z2=Z2F*CTD(KR)
Z3=Z2*RHDP
CLZ2=ALOG(Z2)
CLZ3=ALOG(Z3)
D5=280 J=1.05
250 IF (XNUU(I,J,0,EQ*0.0) GO TO 260
T6=T6+XNUU(I,J)*ALGJ(J)
260 IF (KUR(I,J,EQ*0) GO TO 270
T7=T7+GJ(J)
270 IF (BETA(I,J,0,EQ*0) GO TO 280
CLNM1C(I)=CLNM1C(I)-BETA(I,J)*(ALGJ(J)+CLZ3*XMJATD(J,KR))
280 CONTINUE
IF (KR*EQ*0) GO TO 300
IF (CLNM1C(I)+LT*20) GO TO 290
LOGC=F*TRUE*
GO TO 310
290 CHI(I)=1.0-EXP(CLNM1C(I))
GO TO 310
300 CHI(I)=C
310 CLNP1(I)=CLNP1(I)+(XNUU(I-1,*EQ*0)*RHPL+T6
IF (00*1)) 320,350,320
320 IF (T7)) 330,340,330
330 CLTB5=ALOG(RHDP*T7)
COMPUTATION OF RADIATIVE LOSS QDPR AND ENERGY TRANSFER TO THE ELECTRON GAS ODPE

1. If (LOGC) go to 380
2. PICH(I) = PII(I) * CHI(I)
3. Go to 390
4. PICH(I) = EXP(CLNPI(I) + CLTBF)
5. Computation of radiative loss QDPR and energy transfer to the electron gas ODPE

- If (N*F0 + 1) go to 460

---

COMPUTATION OF ENERGY TRANSFER TO THE ELECTRON GAS BY ELASTIC COLLISIONS

1. TEQ = TEP ** 3
2. VIC = VIC1 * RHOP * SORT(TEQ) * ALOG(VIC2 * TEQ / RHOP / GJ(1))
3. QELAS = 0
4. VNCC = VNCC * SORT(TEP) * RHOP * PICMEG(TEP)
5. Go to (500, 510, ICHG)
6. Go to 520
7. QELAS = QELAS + VEJ * GC(J)
8. QELAS = 3 * CRA * RHOP * CGI(1) * CTAP * (CT - CTE) * GJ(1) * QELAS

---
QDPE=QDPE+QELAS
PRES=PRES*RHO*CMA/ROBRA
SEN=CRA*(SEN-ALOG(PRES*CM*PFESA)/CM)
CONTINUE
IF (INEQ0*O AND ISW5BNE0) WRITE (6,COOCDMP)
RETURN

FORMAT (//11H ***ICOUNT=.15)
FORMAT (43HONEGATIVE CONCENTRATION ENCOUNTERED IN COMM)
FORMAT (24H0SU2 LESS THAN 0 IN COMM)
END
SUBROUTINE CXSECT (L, J)
SUBROUTINE CXSECT DETERMINES THE CORRESPONDENCE OF SPECIES PAIRS
BETWEEN THE MASTER SPECIES LIST AND THE TRANSPORT CALCULATIONS.
COMMON /TRANS2/ KKQ(100),NNQ(100),ISEQ(100),NNKQ,I(50)
M=L-J
J=I(J)
IF (ISIGN(M,L-J),EQ.,M) GO TO 10
M=L
J=M
10 RETURN
END
SUBROUTINE DERIVS
LOGICAL ERR, FAILED
REAL ACOM(30), ELEMENT(10), HP(20)
DOUBLE PRECISION AA, CAPX, CDI
COMMON AA(22,24), CDI(20,10), CAPX(20)
COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO,
1 C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP,
2 CPT, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMX,
3 CTP, CTPI, CTT, CX, CXB, CXH, CXHADATEST, DBTEST, DELT1,
4 DELT2, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT,
5 PCTEST, PRES, PREA, PRESB, PRESTH, PRHC, RHAP, RNU, RHOB,
6 RMOBAR, RHOC, RHOP, RHPL, RHTR, RDRARR, ROBAR, SCPLG, SDA,
7 SHMC, SHPG, SC, SL, SL64, SM, SU, SU2, SUMG,
8 TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA
COMMON BE(64), BET(20), BLB(31), CAI(64), CAPXTH(20), DER
1 CCPJ(20), CACT(64), CGI(20), CMU(20), CHI(64), CIH(20),
2 CLN1(64), CLN(20), CMW(20), ETAJ(64), ETAJ(20), GJA(20),
3 GJ3(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), QM(20),
4 QQ(64), SAJ(20), SDCHI(64), SENT(20), SJH(20),
5 SKIL(20), SS(20), TB(30), TFA(22), TFB(20), TFC(20),
6 TFD(20), TFE(20), TFK(20), THE(20), XMJ(20), XNUI(64),
7 COMMON BET(64, 20), ELT(10, 20), GEL(10, 20),
8 XNUJ(64, 20), XNUIJP(64, 20), ZOCC, ZOCCP,
COMMON IC, IM, INEQ, INEOV, IP, IRUN, ISC, ISCP,
1 ISM, ISMKR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2 ISX1, ISX2, ISX2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A,
3 ISW5B, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, MI, NIF,
4 NIT, NNN, NNS, NOS, NOT, NTEST
COMMON IQJ(20), IGM(20), ITB(30), KUR(64, 20), LPIJ(20, 10),
COMMON ACOM, ELEMENT, HP
COMMON /BL, DELBL, BLINT, XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SWR, JDIM, IPCINT
COMMON /ERROR/, ERR
COMMON /ERROR/, ERR
COMMON /ERROR/, ERR
COMMON /ERROR/, ERR
1 NSECTU(2), ISLAPER, NPROFL, NPROFL2, NPROF, NBL
COMMON /TNONEQ/, TNE, TNPAR, ENPAR, 25, NT, ITR(25), KTF(25),
1 KTR(25), IC(20), IPA(25)
COMMON /TNOC/, XMJATD(20, 2), CLNTD(20), CT(2),
DIMENSION DE(112, 2), DEL2(2), DELO1(2), DELO2(2),
NAMELIST /VHMMP/, ITRY, DELBL, DEL1, DEL2, DEL01, DEL02, DEN
ITRY=0
10 IF (INT.EQ.1) GO TO 30
CTSAVE=CT
CT=CTE
CALL THERM
CALL THERM
CALL THERM
CALL THERM
CALL THERM
CALL THERM
20 XMJATD(1, 2) = XMJATD(1)
CT=CTSAVE
30 CALL THERM
CLNTD(1) = CLNT
CTD(1)=CT
DO 40 I=1,ISS
X*JAT(I,1)=XMJAT(I)
IF (NT*EQ*1) GO TO 50
SHJ(1)=SHJE
CCPJ(1)=CPJE
SENT(1)=SENTE
CTD(2)=CTE
CALL COMM
IF (ERR*OR*FAILED) RETURN
CALL EXACT
IF (ERR) RETURN
IF (DLCGAE*C+6E-0.01) GO TO 60
FAILED=TRUE
RETURN
IF (ISW3B*EQ*0*CR*IPD*EQ*1) GO TO 140
DO 70 K=1,NPRFLS
DELI2(K)=DELI1(K)
CALL OLCALL (,FALSE*)
IF (EMS) RETURN
IF (ABS(1.-DELL(1)/DELI1(1))*GT*0.01) GO TO 80
IF (NPRFLS*EQ*1) GO TO 130
IF (ABS1*-DELL(2)/DELI1(2))*LE*0.01) GO TO 130
DO 90 K=1,NPRFLS
DELO2(K)=DELO1(K)
ITRY=ITRY+1
IF (ITRY*EQ*1) GO TO 10
IF (ITRY*GT*6) GO TO 110
DO 100 K=1,NPRFLS
DEN=DELO2(K)-DELO1(K)-DELI2(K)+DELI1(K)
IF (DEN*EQ*0) GO TO 120
DELL(K)=(DELI1(K)*DELO2(K)-DELO1(K)*DELI2(K))/DEN
GO TO 10
WRITE (6,150)
WRITE (6,*DRVMP)
CONTINUE
END
SUBROUTINE DSMSC (A, I, N)
DOUBLE PRECISION A
C
BASED ON SUBROUTINE SIMQ FROM IBM SCIENTIFIC SUBROUTINE PACKAGE
DATA RNAME /6HDSMSOL/
DIMENSION A(22, 24)

L = 0
DO 20 K = 1, I
DO 10 J = 1, I

L = L + 1
LM = L - N * LN
LN1 = LN + 1
IF (K * EQ * 1) A(J, N1) = A(J, M1)
CALL SIMQ (A(1, 1), A(1, N1), I, KS)
IF (KS * EQ * C) GO TO 30
WRITE (E, 50)
CALL DUMP (RNAME)
RETURN

DO 40 J = 1, I
A(J, M1) = A(J, N1)
RETURN

FORMAT (36HOMATRIX OF COEFFICIENTS IS SINGULAR.)
END
SUBROUTINE DUMP (RNAME)
LOGICAL ERR
COMMON /ERROR/ ERR
ERR=.TRUE.
WRITE (6,10) RNAME
RETURN

10 FORMAT (1H1,24H DUMP ROUTINE CALLED BY *A6)
END
COMMON /NDCE/XMJATD(2C,2),CLNTD(2),CTD(2)
COMMON /TNERK/,SOTE,CTEB,DEMAX
COMMON /TNECE/SUMGH,SCPGH,DPOR,DPDE
COMMON /INIGNE/AINI(10,10)
NAMELIST /DMP1/
NAMELIST /DMP2/
NAMELIST /DMP3/
NAMELIST /DMP4/ DELBL,BLINT,WALL,CPWALL,VISROT,H0
WRITE (6,DMP1) WRITE (6,DMP2) WRITE (6,DMP3) WRITE (6,DMP4)
IF (.NOT.,DATAPE) GO TO 20
DO 10 I=1,NRCOUT
10 BACKSPACE ITPUT WRITE (6,30) ITPUT,NRCOUT NRCOUT=0
WRITE (6,40) RETURN
CC
CC
FORMAT (11H0,***TAPE,13,14H BACKSPACED BY,1I,4I,4H RECORDS TO DELE
1TE DATA FROM FAILED CASE)
FORMAT (1I1) END
SUBROUTINE ELCOND (SIGMA, SSIG)
SUBROUTINE ELCOND COMPUTES ELECTRICAL CONDUCTIVITY IN MHOS/CM.
COMMON /TRANS1/ T(3,20,20), ZM2(20)
COMMON /TRANS2/ B(20,2), BR(20,2), A1(2), X(20), DH(20)
COMMON /TRANS3/ N, IELEC, ID1, ID2
SIGMA=0.
IF (IELEC.EQ.0) GO TO 20
DO 10 I=2,N
SIGMA=SIGMA+X(I)*Q(I,1,1)
10 SIGMA=SSIG*X(1)/SIGMA
RETURN
END
SUBROUTINE E 'VE (ET, IP)

C TIMING ROUTI.
C
C ET=ELAPSED TIME FROM BEGINNING OF EXECUTION IN SECONDS
C IP=CONTROL FOR PRINTING THE TIME
   IP = 0      SUPPRESS PRINT

C LOGICAL CALLED
DATA CALLED / *FALSE*/
ETP=ET0
ETO=ET
IF (CALLED) GO TO 10
ETO=0.
CALL RESET
CALLED=True.
ETO=0.
GO TO 30

10 CALL TIME (I)
ET=0.001*;
IF (IP.EQ.0) GO TO 20
ETM=ET/60.
DET=ET-ETO
WRITE (6,10) ETM, DET.
GO TO 30

20 ETO=ETP

30 RETURN

C FORVAT 24H0**********ELAPSED TIME=,F6.2,28H MINUTES SINCE START 0ETL
IF RUN..F5.0,32H SECONDS SINCE LAST PRINTED TIME/
END
SUBROUTINE EPART (IR, TE, EF, ER, OF, OR)
COMMON /TNONEO/ CTE, DTE, ERPAR, EPAR(2, 25), NT, ITR(25), KTF(25),
  KTR(25), LCH(20), IPA(25)
DEFINITIONS OF ARGUMENTS
INPUTS
  IR = INDEX OF REACTIONS
  TE = ELECTRON TEMPERATURE (DEG. K)
OUTPUTS
  ET = ENERGY GAINED BY THE ELECTRON GAS IN THE FORWARD
  REACTION (CAL/MOLE)
  ER = ENERGY GAINED BY THE ELECTRON GAS IN THE REVERSE
  REACTION (CAL/MOLE)
  OF = ENERGY LOST BY RADIATION IN THE FORWARD REACTION (CAL/
  MOLE)
  OR = ENERGY LOST BY RADIATION IN THE REVERSE REACTION (CAL/
  MOLE)
DATA : HR/2.9808/, RO/1.9872/
IT=ITR(IR)
EO=EPAR(1, IR)
GO TO (10, 20, 50, 20, 80, 90, 10)
10 EF=-EPAR(2, IR)*RO*TE
  GO TO 30
20 EF=-TH*TE
  IF (IT*EO=4) GO TO 70
30 ER=EO-EF
  QF=EO-EF
  GO TO 100
40 OR=0
  GO TO 100
50 EF=0
  ER=0
  QF=0
  GO TO 40
60 GO TO 40
70 ER=EF
  GO TO 60
80 EF=EO
  GO TO 70
90 QF=EO
  EF=0
  ER=0
  GO TO 40
100 RETURN
END
SUBROUTINE EOCALC (T,P)
LOGICAL ERR
CALCULATES THERMOCHEMICAL EQUILIBRIUM FOR SPECIFIED T AND P.
T=TEMPERATURE IN DEG. K
P=PRESSURE IN ATM.
REAL ACC(30), ELM(10), HP(20)
DOUBLE PRECISION AA, CAPX, GJ, CI
COMMON AA(27,24), CI(20,10), CAPX(20)
COMMON A, ANTS, AFNX, AMACH, AR, AR6A, AR8B, BZERO
1 C, CAPL, CH, CHA, CLNT, CM, CMA, CRR, CRP
2 CRRD, CRS, CSTA, CT, CTAP, CTB, CTG, CMT, CMXX, CTMAX
3 CTAP, CTPL, CT, CX, CXM, DATEST, DATES, DATES, DEL1
4 DELT2, DELT3, DLORG, DLUG, DT, EMT, FLUX, HDELX, PCT
5 PRET, PRES, PRESB, PREG, PRESH, PRM, RHP, RMO, RMB
6 RHOB, RHOC, RHOM, RRO, RHST, S, SOUT, SS, SSU, SUMG
7 SENG, SPPG, SC, SL, SL6A, SM, SU, SUU, SUMG
8 TEST, TESTB, TPRINT, TSTOP, TSTOP UP, TSTOP ZPA
COMMON B(64), BET(20), BBL(31), CAI(64), CAPXTH(20)
1 CCPJ(20), CEACT(64), CGI(20), CHI(64), CHI(20)
2 CNIM(64), CLNPI(64), CM(29), DIAM(64), DIAM(20)
3 GJ(20), HERTGJ(20), PGJ(20), PI(64), PCHIP(64), GM(20)
4 GQ(64), SJ(20), SDCI(64), SENT(20), SHJ(20), SHJ(20)
5 SKL(20), SSI(20), TH(30), TFA(20), TF(20), TFC(20)
6 TFD(20), TFE(20), TPK(20), TMEV(20), XMJAT(20), XNUI(64)
COMMON BETA(64,22), E(10,20), FLJ(10,20)
1 XNUI(64,20), XNUIJ(64,20)
COMMON IC, IM, INFO, IP, IRUN, ISC, ISCP
1 ISMC, IXMCNR, ISR, ISS, ISSNR, ISSP, ISSP2, ISSP3, ISSP4
2 ISw1A, ISw1B, ISw2, ISw3, ISw4, ISw5, ISw6, ISw7
3 ISw8, ISw9, ISw10, ISW11, ISW12, IUPQ, IZFRQ, JUK, LC, M1, NF1
4 NIT, NNN, NNS, NOS, NOT, NTEST
COMMON IGJ(20), IGM(20), IT(5), KUR(64,20), LPIJ(20,10)
COMMON ACCM, ELMNT, HP
DIMENSION AAA(22,24), BTA(64,20), CAPX(31), CCI(20), DGJ(20), DJ(20), ERC(20)
1 DJ(20), SDGJ(20), SHJAP(20), THEVP(27)
EQUIVALENCE (AA(1,1),AA(1,1)), (BETA(1,1),BTA(1,1)),
1 (ULB(1,1),CAPX(1,1)), (CAPXTH(1,1),CCI(1)), (GJ(1),GJ(1)),
2 (CAPX(1,1),GJ(1)), (ISAJ(1,1)), (ISJJ(1,1)), (SS(1),SDGJ(1))
3 (SHJAP(1,1),SHJAP(1,1)), (THEVP(1))
COMMON /ERRNGL/ ERR
COMMON /EOC/ ZCAPC(20), ZSEN, ZCH, ZCM, ZRBP, ZRHO
COMMON /EOC2/ ZFZ, ZGMU(20)
DATA RNAME /6MEGCALC/
CTSAVE = CT
CT = T / CTAP
DO 10 I=1, ISC
2 ZCAP(1) = QM(I)
1 IF (QM(I) .LE. 0.4) ZCAP(I) = 1.0 - E-3
3 ZGMU(I) = ALOG(ZCAP(I))
4 ZPZ = ALOG(I)
5 CALL THERM
6 IF (ISMC .EQ. 0) GO TO 60
7 DO 30 I = 1, ISC
8 IL = I + ISC
9 CHII(I) = XMJAT(IL) + ZPZ * BET(I)
10 EOC 1
EOC 2
EOC 3
EOC 4
EOC 5
EOC 6
EOC 7
EOC 8
EOC 9
EOC 10
EOC 11
EOC 12
EOC 13
EOC 14
EOC 15
EOC 16
EOC 17
EOC 18
EOC 19
EOC 20
EOC 21
EOC 22
EOC 23
EOC 24
EOC 25
EOC 26
EOC 27
EOC 28
EOC 29
EOC 30
EOC 31
EOC 32
EOC 33
EOC 34
EOC 35
EOC 36
EOC 37
EOC 38
EOC 39
EOC 40
EOC 41
EOC 42
EOC 43
EOC 44
EOC 45
EOC 46
EOC 47
EOC 48
EOC 49
EOC 50
EOC 51
EOC 52
EOC 53
EOC 54
EOC 55
DO 20 J=1,ISC
CHII(I) = CHII(I) + XMJAT(J) * CDIJ(I,J)
20 CONTINUE
DO 50 I=1,ISMC
SKIL(I) = CHII(I)
50 J=1,ISC
SKIL(J) = SKIL(J) + CDIJ(I,J) * ZGMU(J)
L = I + ISC
ZGMU(L) = SKIL(I)
SKIL(J) = EXP(SKIL(J))
50 ZCAP(L) = SKIL(I)
60 M1 = ISC + 1
N = M1 + 1
DO 90 J = 1, ISC
AA(J,M1) = OM(J) - ZCAP(J)
70 IF (ISMC .EQ. 0) GO TO 90
DO 80 I = 1, ISMC
80 AA(J,M1) = -(CDIJ(I,J) - OM(J) * BET(I)) * SKIL(I) + AA(J,M1)
90 CONTINUE
DO 140 J = 1, ISC
100 AA(J,K) = ZCAP(J)
GO TO 120
110 AA(J,K) = 0.0
120 IF (ISMC .EQ. 0) GO TO 140
DO 130 I = 1, ISMC
130 AA(J,K) = AA(J,K) + (CDIJ(I,K) * CDIJ(I,J) - OM(J) * BET(I)) * SKIL(I)
140 CONTINUE
CALL DPKSC (AA, ISC, 22)
150 IF (E3RA) RETURN
DO 160 K = 1, ISC
ZA = 1.0 + AA(K,M1)
160 IF (ZA) 150, 150, 160
150 ZCAP(K) = ZCAP(K) / 2.
GO TO 170
160 ZCAP(K) = ZCAP(K) * ZA
170 ZGMU(K) = ALOG(ZCAP(K))
180 CONTINUE
IF (ISMC .EQ. 0) GO TO 210
DO 200 I = 1, ISMC
SKIL(I) = CHII(I)
200 J = 1, ISC
SKIL(J) = SKIL(J) + CDIJ(I,J) * ZGMU(J)
J = J + ISC
ZGMU(J) = SKIL(I)
SKIL(J) = EXP(SKIL(J))
ZCAP(J) = SKIL(J)
200 CONTINUE
DO 220 K = 1, ISC
IF (ABS(AA(K,M1)) .LE. TEST) GO TO 220
IF (N .NE. TEST) 70, 70, 240
220 CONTINUE
ZCH = 0.
E0C 56
E0C 57
E0C 58
E0C 59
E0C 60
E0C 61
E0C 62
E0C 63
E0C 64
E0C 65
E0C 66
E0C 67
E0C 68
E0C 69
E0C 70
E0C 71
E0C 72
E0C 73
E0C 74
E0C 75
E0C 76
E0C 77
E0C 78
E0C 79
E0C 80
E0C 81
E0C 82
E0C 83
E0C 84
E0C 85
E0C 86
E0C 87
E0C 88
E0C 89
E0C 90
E0C 91
E0C 92
E0C 93
E0C 94
E0C 95
E0C 96
E0C 97
E0C 98
E0C 99
E0C 100
E0C 101
E0C 102
E0C 103
E0C 104
E0C 105
E0C 106
E0C 107
E0C 108
E0C 109
E0C 110
ZSEM=-ZPZ
ZCM=0
DO 230 I=1,ISS
ZSEM=ZSEM+ZCAP(I)*(SENT(I)-ZGMU(I))
ZCM=ZCM+ZCAP(I)*SHJ(I)
230
ZCM=ZCM+ZCAP(I)*CGI(I)
ZHRP=P+ZCM/T/92.056
ZHRP=ZHRP/(1.0+ZRBP*BZERO)
ZCM=ZCM+(ZHRP/ZHRP-1.0)
CT=CTSAVE
RETURN
240
WHITE (6,250)
CALL DUMP (RNAME)
RETURN
C
C
250 FORMAT (3SH1 TO0, MANY NEWTON-RAPHSON ITERATIONS)
END
SUBROUTINE EXACT
LOGICAL ERR
REAL AGCD(30), ELEMENT(10), HP(20)
DOUBLE PRECISION AA, AA+, CAPX, GJ, CDIJ
COMMON AA(22, 24), CDIJ(20, 20), CAPX(20)
COMMON A, ANTS, AFN, AMACH, AR, ARBA, ARRE, BZERO
1 C, CAR, CH, CHAR, CLNT, CM, C.A., CRA, CRP
2 CRRH, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMXA
3 CTP, CTPL, CTT, CT, CX, CMAX, CTMAX, CTMXA
4 DEL2, DELTAX, DLOG, DLOGR, DT, ENT, FLUX, HDELX, PCT
5 EPTEST, PRES, PRESA, PREST, PTH, PRHQ, RHAP, RHQ, RHO
6 RHQHAR, RHOC, RHPL, RHLT, ROBARA, ROBAR, SPCG, SRT
7 SEN, SHPG, SC, SL, SL64, SM, SU, SU2, SUMG
8 TEST, TEBIT, TPRINT, TSTOP, UP, ZP, ZPA
COMMON BE(64), BET(20), BLBK(31), CAI(64), CAPTH(20)
1 CCPJ(20), CEACT(64), CGI(20), CMU(20), CHI(64), CHI(20)
2 CLNMC(64), CLNP(64), CMW(20), ETAI(64), ETAJ(20), GJA(20)
3 GJB(20), PERTGJ(20), PGJ(20), P1(64), PICH(64)
4 G(n4), SAJ(20), SDCHI(64), SENT(20), SHJ(20)
5 SKL(20), SS(20), TB(30), TFA(20), TFB(20), TFC(20)
6 TFCD(20), TFD(20), THEV(20), TI(I)(20), XNUI(64)
COMMON HET(64, 20), ELJ(10, 20), GELH(10, 20)
1 XNU1J(64, 20), XNU1JP(64, 20)
COMMON IC, IM, INEG, INEQ, IP, IRUN, ISC, ISCP1
1 ISMC, ISMCN, ISR, ISSN, ISSN, ISSN, ISSP2, ISSP3
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A
3 ISW5A, ISW6A, ISW6B, ISWD, IUPD, IZERO, JK, LC
4 NIT, NNN, NNS, NO, NIT, NTEST
COMMON IGJ(20), ILMG(20), ITO(5), KUR(64, 20), LPIJ(20, 10)
COMMON ACD, ELEMENT, HP
DIMENSION AAA(22, 24), BTA(64, 20), CAPQ(31), CCII(20), DGJ(20), GJ(20), SBE
1 J1(20), SDGJ(20), SHJAP(20), THEV(20)
2 EQUIVALENCE (AA11, 1), AAA(11), (HEA11), BTA(11)
1 ((LPX(1), CAPQ(1)), (CAXTH(1), TCM(1), GJ(1)), (DJJ(1)), (GJ(1))
2 (CAPX(1), GJ(1)), (SAJ(1), SBJ(1)), (SS(1), SDGJ(1))
3 (SHJ(1), SHJAP(1)), (THEV(1), THEV(1))
COMMON /PDV, SHJID
COMMON /ERR, ERR
COMMON /SS, CAS, US
COMMON /TFMK/ SDTE, CTEB, DCHA, CHG, SDCHA, DMAX, SFAIL
COMMON /TNEQ/ CTE, DTE, BPAR, BPAR(2, 25), NT, IT(25), KTF(25)
1 KTR(25), ICH(20), IP(25)
2 COMMON /TNEP/ SUGM, SCPH, CDPR, ODPE
NAMELIST /EXTMPP/ CT, CMA, SU2, DLOGA, DLOGR, SUGM, SHIDGJ, SUMTDG
1 IUPD, NT, DTE, DTE, DGP, PI, SI, SHJ
1 =SU2/CMA
IF (INEQ, NF = 0) GO TO 100
10 J1 = 3, ISSP4
20 AAA(1, J1)+ EXA 2
10 J1 = 1, ISSP4
20 AAA(1, J1) = LP112(J1, 1)
10 D3 = 1, ISSP4
20 D3 = 1, ISSP4
S1=BET(K)+SHJ(I)/CT
DO 30 J=1,1SC
AAA(1,J)=CDIJ(K,J)/GJ(J)
S1=S1-CDIJ(K,J)*SHJ(J)/CT
DO 50 J=1SCP1,ISS
IF (I-J) 50, 40, 50
AAA(I,J)=1.0*SHJ(J)
50 CONTINUE
AAA(I,1ISP1)=S1/CT
AAA(I,1ISP2)=BET(K)
60 CONTINUE
DO 70 J=1,ISS
AAA(IISP1,J)=1.0
AAA(IISP2,J)=SHJ(J)
AAA(IISP1,IISP1)=SUMG/CT
AAA(IISP1,IISP2)=SUMG-Z1/CT
AAA(IISP1,IISP3)=Z1/CT
AAA(IISP2,IISP1)=SCPG
AAA(IISP2,IISP2)=Z1
AAA(IISP2,IISP3)=Z1
DO 80 I=IISP1,IISP2
AAA(I,1ISP3)=DLOGA*AAA(I,1ISP3)
CALL DMSQD (AAA,1ISP2,22)
IF (EHR) RETURN
DO 90 I=1,ISS
DGJ(I)=AAA(I,1ISP3)
DT=AAA(IISP1,1ISP3)
DGJ=AAA(IISP2,1ISP3)
GO TO 100
70 CONTINUE
NE=NT+1
90 IF (NT.EQ.2) DCHA=-(CMA/CSP/RHAP/US)*QDPR/RHD/SU
SHJDGJ=0
SUMDGJ=0
DGJ(J)=0
DO 120 J=1,ISS
DGJ(J)=0
DO 110 I=1,ISR
DGJ(J)=DGJ(J)*BETA(I,J)*P(ICHI(I)
SHJDGJ=SHJDGJ+SHJ(J)*DGJ(J)
110 IF (J.LT.NT) GO TO 120
SUMDGJ=SUMDGJ+DGJ(J)
120 CONTINUE
SUMDGJ=CT*SUMDGJ
IF (NT.EQ.2) SUMDG=SUMDG+CTE*DGJ(1)
ECQEF=SQRT(CMA/B,31434E7/CTAP)/RHD/SU/CSP
AAA(1,1)=SCPGH
AAA(I,NF+1)=CCPJ(1)*GJ(1)
AAA(I,12)+Z1
90 IF (NT.EQ.1) GO TO 130
AAA(I,NF+1)=Z1*DLOGA-SHJDGJ
130 IF (NT.EQ.1) GO TO 130
AAA(I,4)=AAA(I,4)-ECQEF*QDPR
AAA(I,2)+SUMGH
130 IF (NT.EQ.1) AAA(2,3)=GJ(1)
AAA(2,2)=CT*SUMGH-Z1*(NT-1)*GJ(1)*CTE
AAA(2,2)=CT*SUMGH-Z1*(NT-1)*GJ(1)*CTE
EXA 56
EXA 57
EXA 58
EXA 59
EXA 60
EXA 61
EXA 62
EXA 63
EXA 64
EXA 65
EXA 66
EXA 67
EXA 68
EXA 69
EXA 70
EXA 71
EXA 72
EXA 73
EXA 74
EXA 75
EXA 76
EXA 77
EXA 78
EXA 79
EXA 80
EXA 81
EXA 82
EXA 83
EXA 84
EXA 85
EXA 86
EXA 87
EXA 88
EXA 89
EXA 90
EXA 91
EXA 92
EXA 93
EXA 94
EXA 95
EXA 96
EXA 97
EXA 98
EXA 99
EXA 100
EXA 101
EXA 102
EXA 103
EXA 104
EXA 105
EXA 106
EXA 107
EXA 108
EXA 109
EXA 110
AAA(2, NE+1) = SUMTDG + Z1 * DLOGA

140 IF (NT EQ 1) GO TO 150
AAA(3, 1) = 0
AAA(3, 3) = 1.5 * GJ(1)
IF (1UPD EQ 1) GO TO 150
AAA(3, 2) = -GJ(1) * CTE
AAA(3, 4) = -1.5 * CTE * DGJ(1) + ECOEF * QDPE

150 IF (1UPD EQ 0) GO TO 170
AAA(1, NE+1) = Z1 * DLOGR - SHJDGJ
IF (NT EQ 1) GO TO 160
AAA(1, 4) = AAA(1, 4) - ECOEF * QDPR
AAA(2, 2) = -21
AAA(2, NE+1) = SUMTDG - DLOGR * (CT * SUMGH - Z1 * (NT-1) * GJ(1) * CTE)
AAA(3, 2) = 0
IF (NT EQ 0) GO TO 170
AAA(3, 4) = GJ(1) * CTE * DLOGR - 1.5 * CTE * DGJ(1) + ECOEF * QDPE

170 CALL D3MSOL (AAA, NE, 22)
IF (ERR) RETURN
DT = AAA(1, NE+1)
IF (1UPD EQ 0) DLOGR = AAA(2, NE+1)
IF (1UPD EQ 1) DLOGA = AAA(2, NE+1)
IF (NT EQ 2) DTE = AAA(3, NE+1)

180 IF (DLOGA NE 0) GO TO 190
AMACH = 1
GO TO 200

190 AMACH = 1 / SQRT(ABS(1 + DLOGA / DLOGR))

200 IF (INEQ NE 0 AND ISW5B NE 0) WRITE (6, EXTDP)
RETURN
END
SUBROUTINE FINDX (A, UPDOWN, X)
LOGICAL ERR, UPPER
SOLVES GEOMETRIC AREA RELATION FOR X, GIVEN THE VALUE A(X).
A = GEOMETRIC AREA RATIO (=1.0 AT GEOMETRIC THROAT)
UPDOWN = 1.0 IF DOWNSTREAM SOLUTION IS DESIRED
       = -1.0 IF UPSTREAM SOLUTION IS DESIRED
X = VALUE OF X OBTAINED
ENTRY FINDX SOLVES FOR X AT WHICH THE MBL-TH PROFILE HAS A
HALF-WIDTH OF A.
COMMON /ERROR/, ERR/
DIMENSION Z(2), DZDX(2), RNAME(2)
DATA RNAME(5)=INDEXX, 6HFINDXC/
NAMELIST /DMP/ V, AR, DADX, VO, VL, VU, ERRL, ERRR, ERRU, ERR, UPPER
ENTRY=1
GO TO 10
ENTRY FINDX(A, MBL, X)
ENTRY=2
UPDOWN=1,
N=0
V=100.
GO TO 60
10 V=1,
   IF (ABS(A+1.0) .LT. 1.0E-5) GO TO 40
   IF (A-1.0) .GT. 40, 50
   A=TR(6.170, A
20 CALL DUMP (RNAME(ENTRY))
30 X=0.
   RETURN
50 VL=1.
   ERR=1.-A
   UP=FALSE.
   ERRU=1.*E30
   NEWTON-RAPHSON ITERATION TO SOLVE FOR X
60 N=N+1
   IF (N .LE. 50) GO TO 70
   WRITE (6, 180) A, UPDOWN, IENTRY, MBL
   WRITE (6, 170) DADX
   GO TO 30
70 GO TO (80, 130), IENTRY
80 CALL GEMAR (SIGN(V, UPDOWN), AR, DADX)
   IF (ERR) RETURN
   ERR=AR-A
   IF (ERR) 90, 160, 110
100 IF (ERR) 140, 140, 100
110 GO TO 140
120 IF (ERR) 120, 140, 140
130 CALL GMAR3 (V, DZDX(1), DZDX(2), Z(1), Z(2))
   AR=Z(MBL)
DADX=ODDX(MSL)
140 IF (ABS(1.0-AR/A)*LE.1.E-5) GO TO 160
IF (N.E.10) GO TO 150
IF (ENTRY.EQ.2.OR..NOT._UPPER) GO TO 150
V=VL-(VU-VL)*ERRL/(ERRU-ERRL)
GO TO 60
150 V0=V
V=V*(A-AR)/ABS(CADX)
V=A_MIN(V,2.*V0)
GO TO 60
160 X=SIGN(V,UPDOWN)
RETURN
C
170 FORMAT (4(N0H0,INDEX CALLED WITH AN AREA RATIO LESS THAN UNITY,.10X.2FDX
1HA=,1PE15.8)
180 FORMAT (3(N0H0,MORE THAN 50 ITERATIONS IN FINDX,.10X.2HA=,1PD15.8,.10X.2FDX
17HUPDOWN=.0PF3+0.10X.7HENTRY=,12,.10X.4MMML=,12)
END
C FROZEN FLOW SOLUTION
ENTRY FROZEN
ISOLV=1
C DETERMINE SONIC CONDITIONS IN FROZEN FLOW
DEL = DELT
F1 = 0.0
T1 = 1.0
CT = T1 - DELT
S1 = 0.0
CLNM4 = ALOG(CMA)
10 IC J = 1, ISS
SS(J) = ALOG(GJA(J))
S1 = S1 - GJA(J)*(SS(J) + CLNM4)
S2 = (ZPA/CMA)
EHT = S1 + S2 - CRRB
20 T2 = CT
CALL THERM
CALL PROP
F2 = FLUX
30 CT = CT - DELT
T3 = CT
CALL THERM
CALL PROP
F3 = FLUX
IF (F3 - F2) 40, 60, 80
IF (DELT - TESTB) 90, 50, 50
50 DELT = DELT/2.0
CT = T1 - DELT
GO TO 20
60 IF (DELT - TESTB) 90, 70, 70
70 T1 = T2
F1 = F2
DEL = DELT/2.0
CT = T1 - DELT
GO TO 20
80 T1 = T2
T2 = T3
F1 = F2
F2 = F3
GO TO 30
90 Z = F1 - F2 + F3 - F2
IF (ABS(Z/F2) > 0.00001) GO TO 100
CT = 12
GO TO 110
100 CT=T2*DELT*(F3-F1)/(2.0*Z)
110 CONTINUE
CALL THERM
CALL PROP
S4=FLUX
CTMAX=CT
FRESTH=PRES
RHTH=RHO
IPASS=2
GO TO 120
C EQUILIBRIUM FLOW SOLUTION
ENTRY EQUIL(IPASS)
ISCLN=2
120 INSTRN=1
#SAVE
130 DO 140 J=1,ISS
140 GJ(J)=GJA(J)
IS=INC=1
LASTP=FALSE
ISIP=FALSE
DELT1=DELT
IF (ISW3H*NE.0) DELT1=0.049*(1.-CTMAX)
XMC1=XMP1
ITS=1
STEPC=FALSE
MODLPT=FALSE
CALL NEXTMP (ITS,XMC1,XMODEL)
INEC=C
141 T2E=CTAP
142 DO 150 I=1,13
150 BLVAR(I)=.5
EPSLON=C
T2F=CTAP
EPSF=0
151 DO 150 I=1,NPRFSL
160 W3DST(I)=1.
ISW3H=0
SWICH=FALSE
IPINT=1
PRES=1.0
CT=1.0
RHO=1.0
QH3AP=ROBARA
SU=0.0
AVACH=0.0
AFNA=C.0
RHOD=1.0
PRF03=1.0
DELT=DELT1
XR=2.
XP=3.
FLAG=-1
IF (IPASS*EQ.1) GO TO 180
WRITE (6,630) (SOLN(I,ISCLN),I=1,3)
DO 170 J=1,ISS
170 SAVE(J)=GJA(J)*CMA
A*MACH=0.
AFNX=1.38
VICC=VISCR
SIG=SIGN
CX=-1.E10
CM=CMA
CALL OUT2 ([SCLN]) RETURN
IF (ERR) RETURN
C DECREMENT TEMPERATURE TO GENERATE SOLUTION
10 IF (NOT.TSTEP.C AND (XO-XP).LT.10.) GO TO 190
TSTEP=T.F,"
DELT=AVNI(DELT1,(1.*DELT/((XO-XP)))
DELT=AVNI(DELT1,0.2*DELT1)
190 DELT=AVNI(DELT1,0.1*CT)
CT=CT+DELT
IF (TSONIC.EQ.-1.AND.CT.LE.CTMAX) GO TO 200
IF (CT.LT.TSTOP) GO TO 610
GO TO 220
C THROAT HAS BEEN REACHED. COMPUTE CONDITIONS AT THROAT.
200 IF (ISWB9.AND.0) GO TO 210
FLAG=1
TSONIC=1
GO TO 220
210 CT=CT+DELTV
C=CTMAX
ISUNIC=0
C COMPUTE THE FLOW CONDITIONS FOR TEMPERATURE CT.
220 GO TO (230,240), ISCLN
230 CALL THERM
CALL PERP
GO TO 250
C CALL NEWRAP
IF (ERPA) RETURN
AFNX=AVX1(SM,FLUX,ONE)
250 IF (PRES.GT.0.*PRES) GO TO 260
A*MACH=SU.SORT(RHO*ABS(ALCG(RHO/RAOB)/ALC(G(PRES/PPESB))/PRES)
PRES=PRES
260 AS=SU.*A/S
AS=AS.*AS/CT
DO 290 J=1,ISS
270 GO TO (270,280), ISCLN
SAVE(J)=GJA(J)*CMA
GO TO 290
280 SAVE(J)=GJ(J)*CM
290 CONTINUE
IF (SM.IP) GO TO 360
C DETERMINE X FROM EFFECTIVE AREA RATIO
IF (ISX,EQ.0) GO TO 310
IF (ISCHC) 310,320,330
300 IF (SATX) GO TO 340
IS .CH=ISWTCH+1
IF (ISWTCH.EQ.3) GO TO 330
C X BASED ON APPROXIMATING GEOMETRIC AREA RATIO BY AFNX
310 CALL FINDX (AFNX,FLAG,CX)
   IF (ERR) RETURN
   IF (ISW3B.EQ.0) GO TO 410
   IF (CX.LT.XZERO) GO TO 180
   GO TO 370
320 CX=X0
   DELT1=DELSV
   DELTV=DELT1
   GO TO 370
330 S*TCH=TRUF.*
   XPB=X0
   DXP=DX0-XP
   X CAL C CORRECTED FOR DISPLACEMENT THICKNESS
340 CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)
   IF (ERR) RETURN
   IF (MODLPT) GO TO 360
   IF (CX.GT.XLAST) GO TO 370
   WRITE (6,640)
   IF (IS<3B.EQ.0.OR.AR.GT.4.*AFNX) GO TO 350
   INSTRT=INSTRT+1
   IF (INSTRT.GT.3) GO TO 350
   W=W/2.
   WRITE (6,650) W
   GO TO 130
350 WRITE (6,FROMP)
   RETURN
360 CALL TRANSP (CT*CTAP,PRES*PRESA)
   GO TO 410
370 XLAST=CX
   CALL BLAYER (*TPUE*)
   IF (ERR) RETURN
   IF (IPASS.EQ.1.AND.ISONIC.NE.0) GO TO 180
   IF (ISONIC) 410,330,400
380 DO 390 =1,NPFLS
390 CMDST=CMDSTM
   IF (IP.=.EQ.1) RETURN
   10 CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)
   IF (MODLPT) GO TO 500
   IF (*NOT*SKIP) GO TO 420
   SKIP=.FALSE.
   GO TO 550
420 XP=X0
   X0=CX
   IF (ISW3B.EQ.0) GO TO 440
   DO 430 M=1,NPFLS
   DELRP=M=DELRLC(M)
   430 DELRLO=M=DELRLC(M)
440 IF (CX.LT.XMAX) GO TO 450
   XMODEL=CX.X2
   LAMPT=*TPUE*
450 IF (CX.GE.XMODEL) GO TO 460
   IF (SKIP) GO TO 220
IF (ISONIC NE 0) GO TO 180
CT = CTSTAR
ISONIC = 1
FLAG = 1
GO TO 180

610 IF (NPRFSLE 0.1) GO TO 620
AXISYM = TRUE
CALL MODEL
AXISYM = FALSE
RETURN

620 FORMAT (1H1, 3A4, 9HSOLUTION//)
640 FORMAT (20H0X DECREASED IN FREEZEQ)
650 FORMAT (100H0X RESTART SOLUTION WITH A LARGER AVERAGING DISTANCE FOR)
1 THE BOUNDARY LAYER CORRELATION PARAMETER. NEW W = .F9*6/1H1)
660 FORMAT (73H0X ITERATION TO OBTAIN FREE-STREAM SOLUTION AT MODEL POINT)
IT DID NOT CONVERGE, 5X, 2HX = .1PE10*3, 5X, 7HX MODEL = .E10*3, 3H CM)
END
56 \text{S1} = 1 \\
\text{S2} = 0.5 \\
\text{RHO} = \text{RTH} \\
\text{GO TO 130} \\
60 \text{A2} = \text{S1} \times \text{S1} \\
\text{N} = \text{J} \\
\text{OM} = \text{S1} + 0.5 \times \text{A} \\
\text{IF (S2) 70, 70, 80} \\
70 \text{RHO} = \text{AMN}1*(\text{RTH} \times \text{DEL} \times \text{RHO} \times (1 - C/A2) \times (1/A)) \\
\text{GO TO 90} \\
80 \text{RHO} = \text{AMN}1*(\text{RTH} \times \text{DEL} \times \text{RHO} \times \text{SORT(C)}/\text{S1}) \\
90 \text{N} = \text{N} + 1 \\
\text{IF (N-50) 100, 100, 130} \\
100 \text{CONTINUE} \\
\text{IF (RHO*GE.0) GO TO 110} \\
\text{WRITE (6, 200)} \\
\text{CALL DUMP (RNAME)} \\
\text{RETURN} \\
110 \text{Z} = \text{S1} \times \text{S1} \\
\text{OMZ} = \text{S1} \times \text{Z} \\
\text{AN} = \text{SORT(C/OMZ)} / \text{RHO} \\
\text{DADR} = \text{AN}/\text{RHO} \times (\text{OMZ} \times \text{S1}) / \text{OMZ} \\
\text{DO} = \text{S1} - \text{AN}/\text{DADR} \\
\text{RHO} = \text{RHO+DD} \\
\text{IF (ABS(DD/RHO)*GE.1*0E-6) GO TO 90} \\
\text{Y} = \text{SIGN(16000, CX*(RHO-\text{RTH}})) \\
\text{IF (Y) 130, 130, 120} \\
\text{RHO} = \text{RTH+SIGN(16000, 02, CX)} \\
\text{N} = \text{0} \\
\text{GO TO 90} \\
130 \text{AFNX} = \text{SM}/(\text{SU} \times \text{RHO}) \\
\text{IF (S2+NE.0) GO TO 170} \\
\text{IF (NRFLE.0) GO TO 150} \\
\text{D2LA} = 0. \\
\text{DO 140 J=1, 2} \\
\text{DELSTP}(J) = \text{RO} \times (1 - \text{OMDST}(J)) \\
\text{NSUJ} = \text{NSORTU(J)} \\
\text{D2LA} = \text{D2LA} + 1.0 \times (\text{RTHCM(J)-DELSTP(J)}) \times \text{PARAM(3, NSUJ, J}) \\
\text{GO TO 160} \\
140 \text{NSUJ} = \text{NSORTU(J)} \\
\text{D2LA} = 1.0 / (\text{RTHCM(J)} \times \text{OMDST}(J) \times \text{PARAM(3, NSUJ, J)}) \\
\text{IF (JDIM*NE.0) D2LA=2.0*D2LA/SORTA} \\
\text{GO TO 160} \\
150 \text{DLNCS} = \text{SORT(C*D2LA/A/RTH}} \\
\text{GO TO 160} \\
160 \text{DLOG}=\text{LOG}((2*C)/(A*(S1*RHO)**2-C*(A+2)**)*S2/S1} \\
\text{GO TO 190} \\
170 \text{AFNX} = \text{S1} \\
\text{DLOGA=S2/S1} \\
\text{RHIJ=SM/(SU*AFNX)} \\
190 \text{RETURN} \\
\text{C} \\
\text{C} \\
200 \text{FORMAT (21MONEGATIVE RHO IN GEOM)} \\
\text{END}
FORMAT (40HERROR IN INPUT DATA FOR NOZZLE GEOMETRY, 4H X=, 1PE14.6)
1.9H ARATIC=, E14.6, 9H DERIVA=, E14.6)
END
SUBROUTINE INGAS

LOGICAL ERR

REAL ACOM(30), ELMENT(10), CPX(20)

DOUBLE PRECISION AA, AAI, CAPX, GI, CDIJ, AIN

COMMON AA(22, 24), CDIJ(20, 10), CPX(20)

COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARIB, BZERO

1 C, CARB, CH, CHA, CLNT, CM, CMA, CRAY, CRP

2 CRAY, CTS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMIN

3 CTAP, CTP, CTT, CX, CXI, CXMAX, DATE, DATEST, DELT

4 DELTA, DELTAX, DLOGA, DLOGR, DT, EFT, FLUX, HDELT, PCT

5 DCTEST, DRC, DRECSA, DRCB, DRCST, PRMC, RHA, RHO, RHUB

6 DRCB, DRECSA, DRCST, DRCST, PRMC, RHA, RHO, RHUB

7 SEN, SMPC, SC, SL, SLX, SM, SU, SUZ, SUMT

8 TEST, TEST2, TPRINT, TSTOP, UP, V, ZPA

COMMON BE(64), BFT(20), BLK(31), CALI(64), CAPXTH(20)

1 CCPJ(20), CEA(64), CGI(20), CMU(20), CH(64), CHI(20)

2 CLNTIC(64), CLNPI(64), CMK(20), ETAI(64), ETAJ(20), GJA(20)

3 GJA(20), HRTI(20), PCJ(20), PI(64), PICH(64), QM(20)

4 GQ(64), SAJ(20), SDHI(64), SENT(20), SHJ(20), SHJA(20)

5 SKIL(20), TII(20), TH(30), TIA(20), TFB(20), TFC(20)

6 TFA(20), TFF(20), TFK(20), TMJ(20), TNP(20), TNU(64)

COMMON BETA(64, 20), LLJ(10, 20), GELJ(10, 20)

1 XNU(64, 20), XNUJIP(64, 20)

COMMON IC, IM, INEQ, INEOY, ID, IFUN, ISM, ISCP

1 ISM, ISMCNR, ISR, ISR, ISCSN, ISSP, ISSP2, ISSP3, ISSP4

2 ISKAI, ISKAI, ISKPI, ISKPI, ISW3A, ISW3B, ISW3B, ISW3C, ISW3A

3 ISW3A, ISW3A, ISW5, ISW5, ISW5, ISW5, ISW5

4 NIT, NKN, NNS, NQT, NTEST

COMMON IGJ(20), IGM(20), ITB(5), KUR(64, 20), LPJI(20, 10)

COMMON ACOM, ELMENT, MP

DIMENSION AA(22, 24), BTA(64, 20), CAPO(31), CCI(20), DGI(20), GJI(20), SAPN

12(20), SGOJ(20), SHJAP(20), THEVP(20)

COMMON /EQC/, ZCPX(20), ZSENC1, ZCM, ZRBP, ZRHO

COMMON /EQC2/, ZFZ, ZGMU(20)

COMMON /IGNE/, AIN(10, 10)

COMMON VERNOR, FRK

DO 10 J = 1, ISC

10 DO 10 J = 1, ISC

CALL MATINV (AIN, ISC, 10)

IF (ERR) RETURN

ISMC = 155, ISC

IF (ISMC, EQ, 0) GO TO 60

DO 30 J = 1, ISC

30 CONTINUE

DO 50 I = 1, ISC

50 CONTINUE

CONTINUE
BET(I)=-1.0
DO 40 K=1,ISC
40 BET(I)=BET(I)+CDIJ(I,K)
50 CONTINUE
DO 60 I=1,ISC
60 SUM=0.0
DO 70 K=1,ISC
70 QM(I)=QM(I)+AIN(K,I)*CAPQ(K)
SUM=SUM+QM(I)
CONTINUE
QM(I)=QM(I)/SUM
CCI(I)=0.0
DO 90 J=1,ISC
90 CCI(I)=CCI(I)+LPIJ(I,J)*CMW(J)
DO 100 I=1,ISC
100 CGI(I)=CCI(I)
IF (ISMC.EQ.0) GO TO 130
DO 120 I=1,ISMC
120 IF (ERR) CONTINUE
CONTINUE
ENTRY INTA
CALL EQCALC (CTAP,PFESA)
IF (ERR) RETURN
ZPA=ZPZ
CM=ZCM
SEN=ZSEN
CMA=ZCMA
CM=CM
CRR=SEN/CMA
SEN=CRR*CRA
CRA=CM
DO 140 I=1,ISS
140 CGI(I)=ZGMU(I)
CAPX(I)=ZCAP(I)
CJM=JCAPX(I)/CMA
RDMARP=ZR4P
RHAP=ZR4D
RDMARP=ROPARP/RHAP
RETURN
END
SUBROUTINE INIT
REAL ACCM(30), ELEMNT(10), HP(20)
DOUBLE PRECISION AA, AS, CAPX, J, CDIJ
COMMON AA(22, 24), CDIJ(21, 10), CAPX(20)
COMMON A, AFNTS, APFX, AMACH, AR, ARBA, ARBB, BZERO,
1 C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP,
2 CRRH, CRRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMXX
3 CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT1,
4 DELT1, DELTAX, DLOGX, DT, ENT, FLUX, HOELX, PCT,
5 FOCY, FOCY+, FOCY-, FOCY, FOCY+, FOCY-, FOCYR, FOCYR+, FOCYR-
6 FOCYR-, FOCYR, FOCYR+, FOCYR-, FOCYR, FOCYR+, FOCYR-
7 FOCYR-, FOCYR, FOCYR+, FOCYR-
8 FOCYR, FOCYR, FOCYR+, FOCYR-
9 FOCYR, FOCYR, FOCYR+, FOCYR-
10 FOCYR, FOCYR, FOCYR+, FOCYR-
11 FOCYR, FOCYR, FOCYR+, FOCYR-
12 FOCYR, FOCYR, FOCYR+, FOCYR-
13 FOCYR, FOCYR, FOCYR+, FOCYR-
14 FOCYR, FOCYR, FOCYR+, FOCYR-
15 FOCYR, FOCYR, FOCYR+, FOCYR-
16 FOCYR, FOCYR, FOCYR+, FOCYR-
17 FOCYR, FOCYR, FOCYR+, FOCYR-
18 FOCYR, FOCYR, FOCYR+, FOCYR-
19 FOCYR, FOCYR, FOCYR+, FOCYR-
20 FOCYR, FOCYR, FOCYR+, FOCYR-
21 FOCYR, FOCYR, FOCYR+, FOCYR-
22 FOCYR, FOCYR, FOCYR+, FOCYR-
23 FOCYR, FOCYR, FOCYR+, FOCYR-
24 FOCYR, FOCYR, FOCYR+, FOCYR-
25 FOCYR, FOCYR, FOCYR+, FOCYR-
26 FOCYR, FOCYR, FOCYR+, FOCYR-
27 FOCYR, FOCYR, FOCYR+, FOCYR-
28 FOCYR, FOCYR, FOCYR+, FOCYR-
29 FOCYR, FOCYR, FOCYR+, FOCYR-
30 FOCYR, FOCYR, FOCYR+, FOCYR-
31 FOCYR, FOCYR, FOCYR+, FOCYR-
32 FOCYR, FOCYR, FOCYR+, FOCYR-
33 FOCYR, FOCYR, FOCYR+, FOCYR-
34 FOCYR, FOCYR, FOCYR+, FOCYR-
35 FOCYR, FOCYR, FOCYR+, FOCYR-
36 FOCYR, FOCYR, FOCYR+, FOCYR-
37 FOCYR, FOCYR, FOCYR+, FOCYR-
38 FOCYR, FOCYR, FOCYR+, FOCYR-
39 FOCYR, FOCYR, FOCYR+, FOCYR-
40 FOCYR, FOCYR, FOCYR+, FOCYR-
41 FOCYR, FOCYR, FOCYR+, FOCYR-
42 FOCYR, FOCYR, FOCYR+, FOCYR-
43 FOCYR, FOCYR, FOCYR+, FOCYR-
44 FOCYR, FOCYR, FOCYR+, FOCYR-
45 FOCYR, FOCYR, FOCYR+, FOCYR-
46 FOCYR, FOCYR, FOCYR+, FOCYR-
47 FOCYR, FOCYR, FOCYR+, FOCYR-
48 FOCYR, FOCYR, FOCYR+, FOCYR-
49 FOCYR, FOCYR, FOCYR+, FOCYR-
50 FOCYR, FOCYR, FOCYR+, FOCYR-
51 FOCYR, FOCYR, FOCYR+, FOCYR-
52 FOCYR, FOCYR, FOCYR+, FOCYR-
53 FOCYR, FOCYR, FOCYR+, FOCYR-
54 FOCYR, FOCYR, FOCYR+, FOCYR-
55 FOCYR, FOCYR, FOCYR+, FOCYR-
TEST=C00001
TESTB=C00001
CT=1*0
CTE=1*0
JJK=0
NQT=0
INEQ=0
NTEST=100
INE=1
CRA=1*9872
CPR=CRA CTAP
CTAP=ALCG(CTAP)
IF (KFIT) 10,30,10
T1=CTAP
T2=T1*T1
T3=T1*T2
T4=T1*T3
DO 20 I=1,ISS
IF (10J(I)*EQ.0) GO TO 20
II=15(I)
THEV(I)=T1*SPRP(10,II)
TFC(I)=T2*SPRP(11,II)
TFD(I)=T3*SPRP(12,II)
TFE(I)=T4*SPRP(13,II)
CONTINUE
20 DO 60 J=1,ISS
IF (EJA(J,I)*LE.0.) GO TO 60
II=15(I)
S4J(I)=SPRP(17,II)+(1.5+EJA(J,I))*CTAP
THEV(I)=SPRP(18,II)/CTAP
THEV(I)=THEV(I)
DO 40 J=1,3
THEV(J+1,II)=SPRP(J+40,II)/CTAP
K=15(I)
DO 50 L=1,K
THEV(L,J,I)=SPRP(L,J,II)/CRP
CONTINUE
DO 70 J=1,ISS
II=15(I)
S4J(I)=SPRP(15,II)/CRP
ISMT=ISMC
ISSNP=ISS
RETURN
END
SUBROUTINE KINT (ZKINT)
COMMON /TRANS1/ T*O(3,20,20),ZM2(20)
COMMON /TRANS2/ B(20,2),B+R(20,2),A(2),X(20),DH(20)
COMMON /TRANS3/ N,IELEC,IZ1,IZ2
C
COMPUTE SUM OVER J OF X(J)*DELTA(I,J)
IF (N\#E0\texttimes1) GO TO 20
N1=N+1
DO 10 I=2,N
I1=I-1
NI=N-I1
DO 10 J=1,I1
NJ=N1-J
DO 10 L=1,2
B(I,L)=B(I,L)+X(J)*G(L,J,I)
10
B(NL,L)=B(NL,L)+X(NJ)*G(L,NI,NJ)
C
COMPUTE THERMAL CONDUCTIVITY COMPONENT DUE TO INTERNAL EXCITATION
C
ENERGY
20 ZKINT=0
DO 30 J=1,N
30 ZKINT=ZKINT+DH(I)*X(I)/B(I,L)
RETURN
END
SUBROUTINE LIST

LOGICAL, SUPG, WEDG, AXISYM

REAL, AIOM, 3GCM, ELEMENT(101), MP(20)

DOUBLE PRECISION, 4GCM, CAPX, GJ, CDIJ

COMMON, AA(21, 24), CDIJ(22010), CAPX(20)

COMMON, AFF, AFNX, AMACH, AR, ARBA, ARBB, BZERO

COMMON, CARB, CH, CHA, CLNT, CM, CHA, CRA, CRP

COMMON, CTP, CTPL, CT, CTR, CTMAX, CTMAX

COMMON, DELT2, DLTAX, DLGA, DLGQ, DT, EN, EU, F, FRA, HDELX, PCT

COMMON, PSI, PREA, PRES, PREST, PRAP, RAP, RAP, RDBH

COMMON, N, RNPAN, RNPBAR, RNPBAR, SC, SOT

COMMON, SP, SHPG, SC, SL, SL64, SM, SU, SV, SUNG

COMMON, T, TEST, TPRINT, TSTOP, UP, ZP, ZPA

COMMON, BE(64), BET(20), BBLK(31), CATI(64), CAPX, (20), LIS 15

COMMON, BPJ(20), CEACT(64), CGI(20), CMU(22), CHI(20), LIS 16

COMMON, CHNW(64), CLNP(64), CHW(26), ETAI(64), ETAJ(20), LIS 17

COMMON, OI(20), OITGJ(20), OJ(20), PI(64), PICH(64), OI(20), LIS 18

COMMON, OI(64), SA(20), SDCHI(64), SENT(20), SHJ(20), SHJ(20), LIS 19

COMMON, SKIL(20), SS(30), TD(30), TFA(20), TFB(20), TFC(20), LIS 20

COMMON, TFD(20), TFH(20), TFP(20), THEV(20), XMJAT(20), XNU(64), LIS 21

COMMON, BETA(64), ELJ(10, 29), GELJ(10, 29), LIS 22

COMMON, IC, IM, INFG, IP, IRUN, IS, ISC, ISP1, LIS 23

COMMON, IS, IS, ISS, ISSR, LSSP, LSSP, LSSP, LIS 24

COMMON, IS, IS, ISW2A, ISW3, ISW3, ISW4, ISW4, LIS 25

COMMON, IS, ISW5, ISW6, JU, JU, JU, LIS 26

COMMON, LNG, LNG, LNG, LNG, NOT, NTEST, LIS 27

COMMON, IGJ(20), IGM(20), ITB(15), KUR(64, 20), LIS 28

COMMON, ACOMELEMENT, MP LIS 29

DIMENSION, AAA(22, 24), RTA(64, 20), CAP0(31), CCI(20), D'I(20), SBL LIS 30

COMMON, J(20), SDGJ(20), SHJAP(20), LIS 31

COMMON, EQUivalence (AA(1, 1), AAA(1, 1), (BETA(1, 1), RTA(1, 1)), LIS 32

COMMON, (BLK(1, 1), CAPX(1, 1), (CAPX(1, 1), (CCI(1, 1), (GJ(1, 1)), GJ(1, 1)), LIS 33

COMMON, (CAPX(1, 1), (GJ(1, 1), (GJ(1, 1), (GJ(1, 1), (GJ(1, 1), (GJ(1, 1)), LIS 34

COMMON, (GJ(1, 1), (GJ(1, 1), (GJ(1, 1), (GJ(1, 1), (GJ(1, 1), (GJ(1, 1), LIS 35

COMMON, (SHJ(1, 1), SHJAP(1, 1), THEV(1, 1), THEV(1, 1), LIS 36

COMMON, /MODPAR, /DMPAR, /STAG, /CATFAC, /TMODEL, /XMDP1, /XMDP2 LIS 37

COMMON, /TLPAR, /KDL1M LIS 38

COMMON, /AREA, /ATPI(11, 2), /PARAM(3, 12, 2), /RTHCM(1), /NSEC(2), LIS 39

COMMON, /NSEC(2), /ISHAPE(12, 2), /NPLOL(2), /MFL5, /NBL LIS 40

COMMON, /ROMAIN, /HS, /SUPG, /MK, /FITER LIS 41

COMMON, /MDP, /TSOILM(20), /TSAR(20), /NTS, /NOL LIS 42

COMMON, /NEWMP, /FACMP, /NMODP LIS 43

COMMON, /RDLIST, /IGAS, /IGASH, /NDOZ, /ICHAO LIS 44

COMMON, /WORD, /3D, /HIGH, /HIGH, /HUMET, /HUMET, /MHEM, /SHM, /SHM, /SHEMET, /SHEMET LIS 45

COMMON, /ISAM, /JAM, /JMOL LIS 46

COMMON, /HASL, /SMASL, /SMASS, /TMOXI, /TSTOPI, /IS(20), LIS 47

COMMON, /BC, /DELUS(2), /BLINT(2), /XZERO, /ZWALL, /CPWALL, /VISO, /DIAM(2), LIS 48

COMMON, /S, /ADV, /DIM, /IPCINT LIS 49

COMMON, /NEWFNE, /GAMIN, /NTS, /TTEST, /TTEST, /TTEST, /TTEST, /TTEST, /TTEST LIS 50

COMMON, /POLYAT, /THEV(4, 20), LIS 51

COMMON, /ROB, /ANGLE, /RAD, /RAF, /ROX, /ROX(20), /TWE, /WKE, LIS 52

COMMON, /NRX, /NANGLE, /WEDG, /AXISYM, /ISW9B LIS 53
DJ 140 I=1,1SR
150 WRITE (6,570) I,KTF(I),KTR(I),ITR(I),EPAR(1,1),EPAR(2,1),IPA(I)
160 IF (NFF1-160) WRITE (6,510)
170 IF (I.GT(J)) 180,180,170
180 CONTINUE
190 IF (INEQV,NE,0) WRITE (6,580)
200 WRITE (6,590) HP(J),TFJ(J),TFB(J),TFD(J),TFE(J),TFK(J),SHJAL
210 CONTINUE
220 IF (KMODP1.GE.9.E19.AND.NTS.EQ.0) GO TO 340
230 WRITE (6,600) CKMAXI=CKMAXI/2.E4
240 IF (KMODP1.LT.9.E19) WRITE (6,630) NMODPT,KMODP1,CKMAXI
250 IF (NTS.EQ.0) GO TO 240
260 WRITE (6,640)
270 WRITE (6,650)
280 WRITE (6,660)
290 WRITE (6,670)
300 WRITE (6,680) KMODP1=KMODP1+1
310 IF (BMODP1.EQ.1) GO TO 340
320 WRITE (6,700) ANGLE(I,1),I=1,NANGLE
330 IF (BTMODP1.EQ.1) GO TO 300
340 WRITE (6,720) NWX,DWX,WX1
350 IF (WX(I,1),GT,1.E19,1) GO TO 330
360 FORMAT (1H,99H,ATOMIC FRACTION,ELEMENT,MOLECULAR WEIGTHS)
370 FORMAT (1H,31X,ATOMIC FRACTION,15X,1PE14.6,14X,E14.6)
<table>
<thead>
<tr>
<th>FORMAT</th>
<th>FUNCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1H, 122M)</td>
<td>REACTION NO.</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>CONSTANT FACTOR AI</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>TEMPERATURE</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 166</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>REACTION NO.</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>NU PRIME MATRIX</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 167</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>2Y MATRIX</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 168</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 169</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 170</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 171</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 172</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 173</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 174</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 175</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 176</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 177</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 178</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 179</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 180</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 181</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 182</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 183</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 184</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 185</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 186</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 187</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 188</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 189</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 190</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 191</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 192</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 193</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 194</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 195</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 196</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 197</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 198</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 199</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 200</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 201</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 202</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 203</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 204</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 205</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 206</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 207</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 208</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 209</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 210</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 211</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 212</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 213</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 214</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 215</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 216</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 217</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 218</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 219</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>FORMATTED DATA</td>
</tr>
<tr>
<td>(1H, 122M)</td>
<td>LIS 220</td>
</tr>
</tbody>
</table>
SUBROUTINE MATINV (A, MM, NN)
MATRIX INVERSION
INVERSE REPLACES ORIGINAL MATRIX
CALLING SEQUENCE  
A = ORIGINAL MATRIX  
MM = SIZE TO BE INVERTED  
NN = FIRST SUBSCRIPT OF A IN DIMENSION  
STATEMENT OF MAIN PROGRAM

DOUBLE PRECISION A
DATA RNAME / 'MATINV'
DIMENSION A(NN, 10), IV(10)
M = MM
DO 10 I = 1, M
  10 IV(I) = I
JA = 1
20 JJ = JA
  K = JJ
  K1 = K + 1
BIG = ABS(A(K, K))
  IF (K1) .EQ. 1, STOP
  GO TO 30
30 DO 50 I = K1, M
  50 IF (BIG .GE. ABS(A(I, JJ))) 40, 50, 10
40 K = 1
  BIG = ABS(A(I, JJ))
  CONTINUE
  60 BIG = A(K, JJ)
  IF (BIG) 90, 70, 80
  CONTINUE
  WRITE (6, 300)
  CALL DUMP (RNAME)
  RETURN
80 IF (K .GT. JJ) 90, 110, 90
  J = IV(K)
  IV(K) = IV(JJ)
  IV(JJ) = J
  DO 100 J = 1, M
    B = A(JJ, J)
    A(JJ, J) = A(K, J)
  100 A(K, J) = B
  A(JJ, J) = 1.0
  JA = JJ + 1
  DO 120 J = 1, M
    120 A(JJ, J) = A(JJ, J) / BIG
    JB = J - 1
  IF (JB) 170, 170, 130
  DO 160 I = 1, JB
    BB = A(I, JJ)
    A(I, JJ) = 0.0
    A(I, JJ) = BB
  160 IF (HH) 140, 160, 140
  DO 150 J = 1, M
  150 A(I, J) = A(I, J) - BB * A(JJ, J)
  CONTINUE
  170 IF (JA - M) 180, 180, 220
  DO 210 I = JA, M
    BB = A(I, JJ)
    A(I, JJ) = 0.0
  210 CONTINUE
IF (BB) 190,210,190
190 DO 200 J=1,M
200 A(I,J)=A(I,J)-BB*A(JJ,J)
210 CONTINUE
GO TO 20
220 M1=M-1
230 DO 290,290,230
240 IF (M1) 290,290,230
250 DC 280, J=1, M1
260 IF (IV(I)-J) 250,240,250
270 K=T
280 CONTINUE
WRITE (6,310)
CALL DUMP (RNAME)
RETURN
290 IF (K) IV(J)
300 DO 270, L=1,M
310 AA=A(L,J)
A(L,J)=A(L,K)
A(L,K)=AA
270 CONTINUE
RETURN
300 FORMAT (1H1,26H MATINV, MATRIX SINGULAR /)
310 FORMAT (1H1,37HINDEXING OR STORAGE FAILURE IN MATINV/)
END
EPS=SMV(2.\ ISOLN)
T2=SHV(1.\ ISOLN)
ICOUNT=0
LO=.FALSE.
HI=.FALSE.
50 IF (ISOLN.EQ.1) GO TO 70
DO 60 I=1,ISS
60 SAVEC(I)=GJ(I)*CM1
GO TO 110
70 P2=P1*(J12*(1.\ -EPS))
80 ICOUNT=ICOUNT+1
IF (ICOUNT.GT.30) GO TO 450
CALL EOCALC(T2,P2)
IF (F4R) RETURN
EPSOLD=EPS
EPS=ABS(EPS/EPSCLD-1.)*LE0.001 GO TO 130
ALPHA=P2/ZRHO
DISC=PT**2-PC*ALPHA
IF (DISC.GE.0.) GO TO 100
EPS=EPSOLD
IF (LC) GO TO 90
T2=0.9*T2
GO TO 50
90 T2=3.5*(T2+TLO)
GO TO 50
100 P2=0.5*(P+SQRT(DISC))
GO TO 90
110 ICOUNT=ICOUNT+1
CT=T2/CTAP
IF (ICOUNT.GT.30) GO TO 450
CALL THERM
ZCM=CM1
ZCH=0.
DO 120 I=1,ISS
ZCM=ZCM+SAVEC(I)*SHJ(I)
OP=P1*T2/T1
DISC=PT**2-4.*R1U12*OP
IF (DISC.LT.0.) GO TO 450
P2=0.5*(P+SQRT(DISC))
R2=R1=P2/OP
EPS=R1/R2
130 H=HS-HD*EPS**2
ZCH=ZCH+ZMA/ZCM
IF (ABS(ZCH/H2-1.)*LE0.001) GO TO 170
E2=ZCH+H2
IF (ERZ) 140,140,150
140 ERR=FRZ
TLO=T2
LO=.TRUE.
IF (HI) GO TO 160
T2=1.1*T2
GO TO 50
150 ERRH=ERZ
TH1=T2
GO TO 300
SOD(1)=1/(2./EPS/((1.+0.5*EPS)*ALOG(1.333333/EPS)-0.5*EPS)-1.)
SOD(2)=-1.23*SOTE*ALOG(0.79*SOTE)
MOD 221
290 VGP(2)=0.64*EPS
MOD 222
300 SHVL(1.ISOLN)='T2
MOD 223
SHVL(2.ISOLN)='EPS
MOD 224
IF (*NOT.*AXISYM OR (*ISOLN.EQ.2.*AND.*FSTAG.EQ.0.)) GO TO 440
MOD 225
IF (*ISOLN.EQ.2.) GO TO 320
MOD 226
DO 310 I=1,ISSS
MOD 227
310 SAVEC(I)=ZCAP(I)
MOD 228
HCF=0.
MOD 229
IF (!NOTRAN) GO TO 360
MOD 230
DO 340 I=1,ISSS
MOD 231
320 I=1,ISC
MOD 232
IF (LPIJ(I,J)*EC.*J) GO TO 330
MOD 233
IFE(LPIJ(I,J).*NE.1) GO TO 340
MOD 234
NEL=NEL+1
MOD 235
IF (!NEL.GT.1) GO TO 340
MOD 236
330 CONTINUE
MOD 237
HCF=HCF+SAVEC(I)*SHJA(I)
MOD 238
340 CONTINUE
MOD 239
HCF=HCF+CRP/ZCM
MOD 240
CALL_TRANS(TS,PS)
MOD 241
IF (ERR) RETURN
MOD 242
HSTAG=CRP/CMA*XCH
MOD 243
HA=CPwALL*TMODEL
MOD 244
HRATIO=HCF/(HSTAG-HW)
MOD 245
IF (LEWIS.EQ.2) HRATIO=0.
MOD 246
QSR=0.0730*SQRT(PS)*(HSTAG-HW)
MOD 247
C 0.375=1.8*0.30417, WHERE FACTOR 1.8 CONVERTS ENTHALPIES FROM CAL/GMD
MOD 248
C
MOD 249
C TO BTU/LB.
MOD 250
350 L=1,2
MOD 251
QS3=3.51/(PR#VR#)*0.3*SQ2F(KDIM+1)*SQRT(ZRHO*VISC*U1*VGP(L))*HSTAGMD
MOD 252
1G-HW.
MOD 253
QFRL(L)=QS*{(1.+FLEVIS**ELN(1.-1.))*HRATIO)
MOD 254
QFRL(L)=QS*{(1.+FLEVIS**ELN(2.-1.))*CATFAC-1.)*HRATIO)
MOD 255
CKS1=(EPS/(1.+EPS))*2*RI*U1/VISC*30.48
MOD 256
360 SCOUT(1)=TS
MOD 257
SCOUT(2)=CF(I)*ZRHO/RHAP
MOD 258
SCOUT(3)=PS
MOD 259
SCOUT(4)=VISC*CF(3)
MOD 260
SCOUT(5)=FLEVIS
MOD 261
SCOUT(6)=Z'FN*CRA/ZCM
MOD 262
SCOUT(7)=ZCM
MOD 263
SCOUT(8)=SIGMA
MOD 264
SCOUT(9)=PR
MOD 265
SCOUT(10)=EPS
MOD 266
SCOUT(11)=QFRL(2)
MOD 267
SCOUT(12)=QFRL(2)
MOD 268
SCOUT(13)=QSR
MOD 269
SCOUT(14)=SOD(2)
MOD 270
SCOUT(15)=SOD(1)
MOD 271
SCOUT(16)=QFRL(1)
MOD 272
IF (!NOTRAN) GO TO 370
MOD 273
370 CONTINUE
MOD 274
SUBROUTINE QGRF
SCOUT(17)=QGRF(1.
SCOUT(18)=CNS.
SCOUT(19)=HRATIC
IF (ISCLN*EQ*2) GO TO 390
HCON=CRF/CMA
HE=HCON*CH
HR=HE+SCRT(PR)*HCON*(CHA-CH)
HSTAR=HE+SCRT(PR)*HCON*(CHA-CH)
RSTAR=CTAP*HSTAR/HS/HC
RHOSTR=RHOF*RHAP/TSTAR*CTAP*CT
CALL TRANP (TSTAR,PRES,PRES)
IF (E*F) RETURN
SCOUT(20)=222/PR**0.667*SCUT(RHOSTR*VISCL/V1)*(HR-HW)
GO TO 390
430 IF (TON/EQ) GO TO 390
60 DO 138 C N=11,20
390 WRITE (6,510) (TITLE(I,ISOLN),I=1,3)
LMSC=LNSCC(ISOLN)
WRITE (6,520) (QUANT(I,ISOLN),EQUAL,S,SCOUT(I),I=1,LMSC)
IF (ISOLN*EQ*2) (ISCLN=6B,EQ*0) GO TO 420
WRITE (6,530) (TITLE(I,ISOLN)),I=1,3)
IF (IC*F-EQ) GO TO 400
EMF=SAVEC(1)
SAVEC(I)=EMF*ZRH0/RHAP*TB7/ZCM
400 ISSSP1=(ISS-1)/E+1
400 K=1,ISSS1
LIM1=S*(K-1)+1
LIM2=MINO(S*K,ISS)
WRITE (6,540) (P(I),SAVEC(I),I=LIM1,LIM2)
IF (IC*F-EQ) GO TO 420
SAVEC(I)=EMF
420 IF (TON/EQ) GO TO 430
NRCOUT=NRCOUT+1
IMF=IMP+1
ITYPER=3
WRITE (1POUT,ITYPER,IMP,SCOUT(3),SCOUT(1),SCOUT(11),SCOUT(16),SCMOD
320 OUT(20),SCOUT(19),SCOUT(5),SCOUT(9),SCOUT(4),SCOUT(2)
430 IF (FR*G*EQ*2) AND. NOT. WEDGEM) GO TO 460
440 CONTINUE
GO TO 460
450 WRITE (6,550)
WRITE (6,MODMMP)
460 DO 470 C=CTSAVE
470 IF (NOT(R)) GO TO 470
470 IF (WEDGEM) CALL WEDGE
480 RETURN
FORMAT (68HOMODEL PARAMETER ROUTINE CALLED FOR A MACH NUMBER LESS MOD 329
ITHAN 1.6$

FROZEN PRANDTL NUMBER AT STAGNATION-POINT

1) MOD

TEMPERATURE ON MODEL =$F7.4$

MOD

MODEL CONDITIONS = .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD

MODEL CONDITIONS - .3A6

MOD
60 NIT=0
M2=ISC+2
ZP=ALOG(PRES)
DO 90 J=IM,ISC
AA(J,M2)=QM(J)+CAPX(J)
IF (NODEP5) GO TO 90
DO 80 I=1,ISMCN
AA(J,M2)=AA(J,M2)-(CDIJ(I,J)-QM(J)+BET(I))*SKIL(I)
80 CONTINUE
AA(M1,M2)=ZP-ZPA
DO 100 L=IM,ISSNR
AA(M1,M2)=AA(M1,M2)+(PGL(I)-CGM(L))*CAPX(L)
100 DO 150 J=IM,ISC
AA(M1,M2)=AA(M1,M2)+(PGL(I)-CGM(L))*CAPX(L)
DO 150 K=IM,ISC
IF (J-K) 120,110,120
110 AA(J,K)=CAPX(J)
GO TO 130
120 AA(J,K)=0,0
130 IF (NODEP5) GO TO 150
DO 140 I=1,ISMCN
AA(J,K)=AA(J,K)+(1-PGL(I)+CGM(I))*CAPX(K)
140 CONTINUE
DO 170 K=IM,ISC
AA(M1,K)=AA(M1,K)+1-PGL(K)+CGM(K)*CAPX(K)
IF (NODEP5) GO TO 170
DO 160 I=1,ISMCN
L=I+ISC
AA(M1,K)=AA(M1,K)+1-PGL(L)+CGM(L)*CAPX(L)
160 CONTINUE
AA(M1,M1)=1,0
IF (NODEP5) GO TO 190
DO 180 I=1,ISMCN
L=I+ISC
AA(M1,M1)=AA(M1,M1)+BET(I)*(1-PGL(L)+CGM(L))*CAPX(L)
180 DO 210 J=IM,ISC
AA(J,M1)=0,0
IF (NODEP5) GO TO 210
DO 200 I=1,ISMCN
L=I+ISC
AA(J,M1)=AA(J,M1)+BET(I)*CAPX(L)*CDIJ(I+J)-QM(J)*BET(I)
200 CONTINUE
IF ((JK)(240,220,240)
220 IF (CAPX(1)-1.0D0-20) 230,230,270
230 JK=1
IMCN=ISMCN-IC
ISSN=ISSNR-IC
NODEP5=ISMCN,E0,0
IM=IM+1
240 AA(J,1)=0,0
DO 250 J=2,M2
250 AA(J,J)=0,0
GO TO 260
260 CAPX(1)=0,0
CALL DSMSOL (AA,M1,22)
IF (ERR) RETURN
NIT=NIT+1
DO 340 K=IM,ISC
ZB=1.+AA(K,M2)
IF (ZB) 280, 280, 310
280 CAPX(K)=CAPX(K)/2.
IF (CAPX(K)) 290, 300, 290
290 CGMU(K)=ALOG(CAPX(K))
GO TO 340
300 WRITE (6,480) K
CGMU(K)=CGMU(K)-0.69314718
GO TO 340
310 CAPX(K)=CAPX(K)*ZB
IF (CAPX(K)) 320, 330, 320
320 CGMU(K)=ALOG(CAPX(K))
GO TO 340
330 WRITE (6,480) K
CGMU(K)=CGMU(K)*ALOG(ZB)
CONTINUE
ZC=1.+AA(M1,M2)
IF (ZC) 350, 350, 360
350 PRES=PRES/2.
IF (PRES) 370, 380, 370
360 PRES=PRES*ZC
IF (PRES) 370, 390, 370
370 ZP=ALOG(PRES)
GO TO 400
380 WRITE (6,490)
ZP=ZP-0.69314718
GO TO 400
390 WRITE (6,490)
ZP=ZP+ALOG(ZC)
400 IF (NODES) GO TO 430
DO 420 I=1,IMCNR
420 SKIL(I)=CHI(I)*BET(I)*(ZP+ZPA)
DO 410 L=IM,ISC
410 SKIL(I)=SKIL(I)+C0IJ(I,L)*CGMU(L)
J=1+ISC
CGMU(J)=SKIL(I)
SKIL(I)=EXP(SKIL(I))
CAPX(J)=SKIL(I)
CONTINUE
430 DO 450 K=IM+M1
IF (ABS(AA(K,M2)) LE TEST) GO TO 450
450 WRITE (6,500)
CALL DUMP (RNAME)
RETURN
460 CONTINUE
CM=0.
10 CM=CM+CAPX(I)*CGI(I)
CALL CECAPX(I)*SFI(J(I)
470-475-
RHOBAR = PRES * CM / (CMA * CT)
RHO = (RHOBAR * RUBARA) / (1.0 + RHOBAR * ROBARP * BZERO)
C = CH**2 (CMA / CM)
CH = CH + (PKFS / RHOBAR) * (RHOBAR * (ROBAR / RHO) - 1.0)
DG = 470.0 * IM * ISSNR

470 GJ(I) = CAPX(I) / CM
SU2 = 2.0 * (CHA - CH)
SU = SQRT(SU2)
FLUX = RHO * SU
RETURN

CCC

480 FORMAT (15H IN NEWRAP, CAPX(*12,3H) = 0)
490 FORMAT (14H IN NEWRAP, P = 0)
500 FORMAT (35H1 TOO MANY NEWTON-RAPHSON ITERATIONS)
END
SUBROUTINE NEXTMP (ITS, XMOD1, XMODEL)

LOGICAL ERR
COMMON /ERROR/ ERR
COMMON /AREA/ ATPL(11, 2), PARAM(3, 12, 2), RTHCM(2), NSECT(2),
1. NSECTU(2), ISHAPE(12, 2), NPRFL(2), NPRFLS, NBL
COMMON /NEWMP/ FACMP, NMODPT
COMMON /MODPT/ TSDIAM(20), TSAR(20), NTS, MBL
COMMON /MODPAR/ XMP1, DXMP, FSTAG, CATFAC, TMODEL, XMODP1, DXMODP,
1. TPLATE, KDIM
DATA ONE /1/
IF (NPRFLS.EQ.1) GO TO 20
IF (ITS.LE.nts) GO TO 10
XMODEL=1*E30
GO TO 50
10 CALL FINDXC (TSAR, ITS, MBL, XMOD2)
IF (ERR) RETURN
GO TO 50
20 IF (ITS.GT.nts) GO TO 30
CALL FINDX (TSAR, ITS, ONE, XMOD2)
IF (ERR) RETURN
IF (XMOD1-XMOD2) 30, 30, 40
30 XMODEL=XMOD1
XMOD1=XMOD1+FACMP
GO TO 50
40 XMODEL=XMOD2
ITS=ITS+1
50 RETURN
END
COMMON /GLIM2/ NOREAC(64)
COMMON /AREA/ ATPI(11,2),PARAM(3,12,2),RTHCH(2),NSECTION(2),
1 NSECTU(2),ISHAPE(12,2),NPROFL(2),NPRFLS,NBL
DATA RNAME /5*NONEQ/
COMMON /MODPAR/ XMP1,DXMP,FSTAG, CATFAC,TMODEL,XMODP1,DXMODP,
1 TPLATE,KDIM
COMMON /STAG/ T2E,EPSON,T2F,EPSP
COMMON /MODP2/ MODLP
COMMON /THRT/ RSA
COMMON /SWITCH/ XPB,DXPB,W
COMMON /TEMPRY/ SAVEC(20)
COMMON /AVG/ SSAVE
COMMON /RDWEDG/ ANGLE(10),RADLE(5),WX1,DWX,WX1(20),TWEDGE,WK,
1 NWX,NANGLE,NRADLE,WEDELM,AXM,SYM,ISWSWG
COMMON /TNONFO/ CTE,DTF,EPAR,EPAR(2,25),NTR,ITR(25),KTF(25),
1 KTR(25),1CH(20),IPA(25)
COMMON /TNERK/ STF,CTEB,DCHA,CHB,SDCHA,DQMAX,IFAIL
COMMON /STEPS/ NSTEPS
COMMON /TNCE/ SUMGH,SCPGH,QDPR,QDPE
COMMON /AEGOM/ SQRTA,S1,S2
COMMON /SS/ CAS,US
DIMENSION DELDP(2),IF(30)
DATA TENTH /10000,/, ONE /1,0/ WRITE (6,180)

C INITIALIZATIONS
W=WSAVE
DQMAX=0.
QEOLC=0.
QDPE=0.
NSTEPS=0
ISTEPS=0
ISUBO=0.
T2E=CTAP
EPSON=0.
T2F=CTAP
EPSP=0.
MODLP=FALSE
XMSET=.FALSE.
XMODEL=1.E30
IRA=0
ALOGT=ALOG(TENTH)
ICOUNT=0
DO 12 I=1,ISR
NOREAC(I)=FALSE.
1
10 IPRINT=I
IS5=IS5*5H
DELSV=DELT1
IF(IS5.EQ.0) DELT1=AMINI(0.049*(1.-CTMAX),DELT1)
HDELX=0.5*DELTAX
DXOLD=DELTAX
DO 20 I=1,NPRFLS
20 DELBL(I)=0.
CCP=1.
CS=SEN
S5=1.*1
SCD=2
ITB(I)=0
NNS=0
NNE=0
ISCPl=ISC+1
ISSP2=ISS+1
ISSP3=ISS+2
ISSP4=ISS+3
INEQ=0
DO 30 I=1,ISR
P(I)=1.0
ZZZ=CAl(I)
CAI(I)=ALOG(ZZZ)-ETA(I)*ALOGT
CAISV(I)=CAI(I)
BET(I)=0.0
DO 30 J=1,ISS
BET(I,J)=XNU1P(I,J)-XNU1J(I,J)
BET(I)=REF(I)+BET(A, J)
30 XNU1(I)=XNU1(I)+XNU1J(I,J)
ZZZ=CMA/CRP
CSTA=.5*ALOG(ZZZ)
SL64=SL/0468
C
COMPUTE AND TEST RANK OF BETA MATRIX
IX=0
I=0
I=1
IF (I-ISS) 50,50,150
50 DO 70 J=12,ISR
IF (BET(J+1)) 60,70,60
60 KA=J
IX=IX+1
GO TO 60
70 CONTINUE
GO TO 40
80 K=KA
DO 90 M=1,ISS
AB=BETA(K,M)
BET(A,K)=BETA(IK,M)
90 BET(I)=AB
I=1
IF (I<IX) 100,100,150
100 DO 110 I=1,IX
CA=BL(I)/BET(I)*I)
IF (CA) 110,140,110
110 DO 130 N=I,ISS
BET(A,N)=BET(A,N)-CA*BET(I,X,N)
130 CONTINUE
140 CONTINUE
GO TO 40
150 IF (IX-ISMIC) 160,170,160
160 WRITE (6,1090)
3  CERROR EXIT NO. 1  ************************************************************NON 166
3  CBETA MATRIX OF INCORRECT RANK  NON 167
3  IERR=1 NON 168
3  GO TO 1000 NON 169
C  RESET BETA MATRIX (DESTROYED DURING RANK CALCULATION) NON 170
170 DO 180 I=1,ISR NON 171
3  DO 180 J=1,ISS NON 172
3  BETA(I,J)=XNUIJP(I,J)-XNUIJ(I,J) NON 173
3  AR=0 NON 174
3  LCR=0 NON 175
3  DELT=DELTI NON 176
3  DO 200 I=1,ISC NON 177
3  CI(I)=0 NON 178
3  DO 190 J=1,ISS NON 179
3  C(I)+CI(I)+GJA(J)*LPIJ(J,I) NON 180
200 CONTINUE NON 181
3  GO TO 220 NON 182
C  END OF INITIALIZATIONS NON 183
210 IF (MODLPT) CALL NE:TMP (ITS,XMOD1,XMODEL) NON 184
3  GO TO 230 NON 185
220 CALL DENIVS NON 186
3  IF (ERR) RETURN NON 187
3  IF (NOT*FAILED) GO TO 230 NON 188
C  ERROR EXIT NO. 2  ************************************************************NON 190
3  IERR=2 NON 191
3  GO TO 1000 NON 192
230 IF (XMSET*OR*AMACH*LT*.5) GO TO 240 NON 193
3  XMSET=.TRUE.* NON 194
3  XMOD1=XMP1 NON 195
3  ITS=1 NON 196
3  CALL NEXTMP (ITS,XMOD1,XMODEL) NON 197
3  IF (IUPD) 250,350,250 NON 198
240 IF (DLOGA) 450,260,260 NON 199
250 IF (CX) 450,450,270 NON 200
260 IF (AR) 300,280,300 NON 201
270 AR=1*X NON 202
3  DO 290 I=1,NPRFLS NON 203
3  OMST(I)=1--DEBLI(I) NON 204
3  IF (AFNX*LT*DATEST) GO TO 450 NON 205
C  SAVE DATA AT SWITCH POINT FOR POSSIBLE RESTART NON 206
3  TB(1)=CX NON 207
3  TB(2)=CT NON 208
3  TB(3)=SC NON 209
3  TB(4)=CTT NON 210
3  TB(5)=DELTAX NON 211
3  TB(6)=DELX NON 212
3  TB(7)=CTE NON 213
3  TB(8)=SCLD NON 214
3  TB(9)=CHA NON 215
3  IB(1)=INEQ NON 216
3  IB(2)=NNS NON 217
3  ITB(3)=NNN NON 218
3  ITO(4)=NRDOUT NON 219
3  NON 220
DO 310 I=1,ISS
TB(I+10)=JG(I)
BLK(I)=RSA
IF (ISW3B.EQ.0) GO TO 340
DO 320 I=1,10
320 BLK(I+1)=BNEAR(I)
BLK(I+12)=XMT
BLK(I+13)=BRF
DO 330 I=1,NPRFLS
330 BLK(I+14)=BLINT(I)
BLK(I+15)=XSN(I)
BLK(I+17)=DEBL(I)
C Switch to downstream region
340 CALL THROAT
IF (ERR) RETURN
GO TO 450
IF (DLOGR) 450,450,360
360 WRITE (6,1100)
370 IF ((AFNX.DATTEST).LT.0.05) GO TO 380
C Error exit No. 3
C POSITIVE DLOGR FAR BEYOND SWITCH POINT
C ERR=3
GO TO 1000
C Switch to downstream region was premature.
C INCREASE DATTEST AND RESTART UPSTREAM SOLUTION AT SWITCH POINT.
380 DATTEST=2.*(DATTEST-1)+1
IUPD=1
AR6=ARR6+1.0
IF (ARR6.LT.ARRB) GO TO 390
C Error exit No. 4
C ERR=4
GO TO 1000
390 TSZ=PRINT
TPRINT=0
C CALL PRTA
IF (ERR) RETURN
ITAPE=NRCOUT-ITB(4)
IF (ITAPE.EQ.0) GO TO 410
C Backspace binary output tape to eliminate records of invalid steps beyond switch point
C DO 400 I=1,ITAPE
400 BACKSPACE ITPOUT
410 TPRINT=TSZ
C Reset data for restart at switch point
C X=TB(1)
C T=TB(2)
C SC=TB(3)
C TTT=TB(4)
C DELTAX=TB(5)
C DELTX=TB(6)
C CT=TH(7)
C SCD=TB(8)
C CMA=TB(9)
C INEQ=ITB(1)
NN5=ITB(2)
NN=ITB(3)
NR=OUT=ITD(4)
DO 420 I=1,15S
420
GJ(I)=TB(I+10)
PSA=BLB(I)
IF (ISW3R.EQ.0) GO TO 220
DO 430 I=1,15
430
BLNEAK(I)=BLB(I+1)
XP=BLB(2)
DXP=BLB(3)
DO 440 I=1,NPRF
DLINT(I)=BLB(I+13)
X5N(I)=BLB(I+15)
DEBL(I)=BLB(I+17)
440
ISMD=30
COPLU=*TRUE*
GO TO 220
450
CONTINUE
IF (INCG) 610,460,610
C PERTURBATION SOLUTION
460
CALL PRT
IF (ERR) RETURN
IF (INP.*EQ.*C .AND.*DELT.EQ.DEL1) DELT=DELT
C DETERMINE MAXIMUM AND MINIMUM PERTURBATIONS IN CHI(I)
470
DCHMAX=C
DCHMIN=1.E-30
DO 480 I=1,15
DCH=ABS(DCHI(I))
IF (DCH=DCHMAX) 480,480,470
DCHMAX=DCH
DCHMIN=DCH
IMAX=I
480
F=(DCHMIN.GT.*DCHMAX .AND.*IMAX.GT.*DCH MAX .IMAX
WRITE (6,1110) DCHMIN,DCHMAX,IMAX
DCHMAX=0
DCHMIN=1.E30
DO 480 I=1,15
ADCH=ABS(DCHI(I))
IF (ADCH=DCHMAX) 480,480,470
DCHMAX=ADCH
DCHMIN=ADCH
IMAX=I
490
RATIO=ADH.MAX/ADCH.RAT
IF (RATIO.GE.1) GO TO 510
C EXCESSIVELY SMALL RATIO OF MINIMUM AND MAXIMUM PERTURBATIONS
500
IN CHI(I)
DO 500 J=1,15
IF (BETA(J).LT.PCHI(IMAX).GE.0.) GO TO 500
IF (G(J).GT.GAMIN) GO TO 500
IF (IRA.LE.50) GO TO 490
WRITE (6,1010)
500
GO TO 500
C ARTIFICIAL INCREASE OF RATE CONSTANT TO PREVENT PREMATURE START
C OF INTEGRATION
490
RATIO2=1.1*DCHRAT/RATIO
CA(I)=CA(I)+ALOG(RATIO2)
WRITE (6,1020) RATIO,HPI(J),IMAX,G(J),IMAX,RATIO2
IRA=IRA+1
GO TO 220
500
CONTINUE
C TESTS FOR SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION
IF (DCMAX.LE.CCHI) GO TO 580  
IF (DCMAX.LE.PCTEST+CCHI) GO TO 540  
C  
MAXIMUM DCCHI(I) IS TOO LARGE -- BACKSTEP THE PERTURBATION  
C  
SOLUTION TO FIND THE CORRECT SWITCHING POINT  
C  
DELT=DELT/2.0  
ISUBD=ISUBD+1  
IF (ISUBD.GT.23) GO TO 520  
CT=CT+DELT  
CTE=CT  
NAMELIST NEQDMP  
INEQ.JUPD.DLOGA.AR.CX.CT.DELT.AFNTS.  
C  
IF (ISWB.NE.0) WRITE (6,NEQDMP)  
GO TO 600  
C  
WRITE (6,1120) ISUBD  
WRITE (6,NEQDMP)  
C  
ERROR EXIT NO. 5  
C  
GO TO 1000  
C  
SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION  
C  
DXM=0.01*DCHMIN/DCHLL  
IF (DELTAX.LE.DXM) GO TO 550  
DELTAX=DXM  
HDEL=J5*DELTAX  
TB(5)=DELTAX  
TB(6)=HDELX  
DXOLD=DELTAX  
C  
CT=CT+PXCT  
CTE=CT  
RHQ=RHQ+PRHO  
GO 560 I=1,ISR  
IF (CAI(I).EQ.CAISV(I)) GO TO 560  
CAI(I)=CAISV(I)  
WRITE (6,103C) I  
C  
CONTINUE  
GO 570 J=1,ISS  
GJ(J)=GJ(J)+PERTGJ(J)  
CALL DERIV  
IF (ERR) RETURN  
IF (*NOT*FAILED) GO TO 610  
C  
ERROR EXIT NO. 6  
C  
GO TO 1000  
C  
CONTINUE PERTURBATION SOLUTION  
C  
IF (CX.LE.XMODEL) GO TO 590  
MOOLPT=.TRUE.  
CALL NEXTMP (ITS,XMOD1,XMODEL)  
C  
PRINT POINT IN PERTURBATION SOLUTION  
C  
CALL DATA  
IF (ERR) RETURN  
IF (CT.NE.CXMAX) GO TO 660  
IF (CT.LE.TSTCP) GO TO 660  
ISUBJ=J  
CT=CT+DELT  
CTE=CT
CALL NEWRAP
IF (ERR) RETURN
AFNTS=SM/FLUX
CALL AXFIT
IF (ERR) RETURN
GO TO 220
C  NUMERICAL INTEGRATION
410 IF (CT-.0) 630, 630, 620
520 WRITE (6, 1130)
C  ERROR EXIT NO. 7 ****************************
IF (ERR) RETURN
GO TO 1000
630 IF (CXGE.CXMAX) GO TO 660
IF (CTLE.TSTOP) GO TO 660
NNN=NNN+1
PRINT POINT IN SOLUTION BY INTEGRATION
CALL PRINT
IF (ERR) RETURN
IF (NNN-QNS) 670, 640, 640
640 DELTAX=SC@DELTAX
MDELX=DELTAX/2.0
NNS=NNS+1
NNN=0
GO TO 670
650 SC=SC+1
SCX=SC
NNX=0
FIND POINT
660 TPRINT=0.0
CALL DATA
AXISYM=TRUE
CALL MODEL
WRITE (6, 1160) ISTEPS
C  MAIN RETURN STATEMENT*************************************************
C  RETURN
C  SAVE DATA AT BEGINNING OF INTEGRATION STEP FOR POSSIBLE RESTART
760 DO 660 1=1, 1SS
660 GJH(I)=GJH(I)
C  CONTINUE
C830 CONTINUE
CXH=CX
CTJ=CT
C632 CONTINUE
CARD=AFNX
RHO=RHO
CRS=CRS
CH=NCH
DELX(1)=DELX(1)
IF (QNS/.5E0.2) DELX(2)=DELX(2)
490 IF (CX+DELTAX.LT.XMODEL) GO TO 790
Y1C=ITY1
Y savings=DELX
H0XRAVE=MODELX
DELTA = XMODEL - CX
MDEL = 0.5 * DELTA

700 IF (CX + DELTA * LT * CXMAX) GO TO 710
DELTA = CXMAX - CX
MDEL = 0.5 * DELTA

C INTEGRATION STEP
710 CALL RK1
IF (ERR) RETURN
IF (FAILED) GO TO 740
IF (.NCT * MODLPT) GO TO 720
DELTA = DXSAVE
MDEL = DXSAV

720 CONTINUE
C TESTS FOR VALID INTEGRATION STEP
IFAIL = 1
IF (ABS(SDT/CT) * GT * TTEST) GO TO 740
IF (NT * EQ * 1) GO TO 730
IFAIL = 2
IF (ABS(SDIST/CTE) * GT * TTEST) GO TO 740
IFAIL = 3
IF (ABS(SC/CHA) <= TTEST) GO TO 740
IFAIL = 4
IF (.LOGA * CX * GE * 0.01) GO TO 760
C INTEGRATION STEP FAILED. REDUCE STEP SIZE AND RESTART AT BEGINNING OF STEP
740 DELTA = DELTA / SCD
MDEL = DELTA / 2 * 0
MODLPT = FALSE
ICOUNT = ICOUNT + 1
IF (ICOUNT GT 30 OR DELTA LE 1 E -10) GO TO 990
IF (ICOUNT LT 1) IFAIL = 741
DO 750 I = 1, ISS
750 GJ(I) = GJR(I)
CT = CT
C N = CTEB
CX = CX
AFN = CARB
RHOB = RHOB
SEN = CRS
CHA = CB
DELRL(1) = DELRLB(1)
IF (NNRFLS * EQ * 2) DELBL(2) = DELBLB(2)
NNN = 0
NN = 0
SC = SC - 0.1
SC = MAX(1; SC, 1.1)
SCD = 1.1 * SCD
CALL DERIVS
IF (ERR) RETURN
IF (.NOT. FAILED) GO TO 690
C ERROR. EXIT NO. 8
760 DD 790 I = 1, ISS
Caoa, -
i,.
FAIL=I+10
GJ(I)=GJB(I)+SDGJ(I)
IF (GJ(I)<L+O) GO TO 770
IF (ABS(GJ(I)/GJB(I))>LEGEST) GO TO 790
C SUPPRESS REACTIONS INVOLVING STEP-SIZE CONTROLLING SPECIES
NON 496
GJB(I)=GJ(I)+GAMIN GO TO 740
NON 500
C TEST ONTE TRATION BELOW GAMIN AND FALLING
NSUPR=0
DO 780 K=1,1SR
IF (GJ(I)<O) GO TO 730
NOREAC(K)=TRUE
NSUPR=NSUPR+1
C WRITE OUTPUT DATA
WRITE (6,1053) HP(I),GJB(I),GAMIN,NSUPR
IF (GJ(I)<O) GO TO 790
NON 498
SUPP=NSUPR
C VALID STEP, APPLY INCREMENTS TO DEPENDENT VARIABLES.
NON 500
DO 810 I=1,1SR
C =CTB+SDT
CTE=CTE+SDLT
C =CHS+SDCHA
CHF=CHF+SDCHA
NON 502
C RESTORE CONSERVATION OF CHEMICAL ELEMENTS
NON 504
DO 850 J=1,1SR
C =CJ+GJ(I)*LPIJ(I,J)
C =CJ+GJ(I)*LPIJ(I,J)
NON 506
DO 880 I=1,1SR
GSQ(I)=GJ(I)**2
NON 508
IF (GSQ(I)) OR (LPH*ISC) GO TO 830
C
NON 509
WRITE (6,1050) I
C Go to 740
NON 511
C ERROR EXIT NO. 9
YER=9
GO TO 1000
NON 512
C
NON 513
DO 880 I=1,1SR
C =ISMP1=ISMC+1
ISMP1=ISMC+1
NON 514
DO 850 K=1,1SR
DCA(K)=0.
NON 515
DO 860 J=1,1SR
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 516
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 517
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 518
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 519
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 520
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 521
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 522
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 523
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 524
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 525
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 526
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 527
WRITE (6,1060) I
NON 528
C Go to 1000
NON 529
YER=9
GO TO 1000
NON 530
C
NON 531
DO 850 K=1,1SR
DCA(K)=0.
NON 532
DO 860 J=1,1SR
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 533
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 534
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 535
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 536
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 537
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 538
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 539
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 540
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 541
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 542
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 543
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 544
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 545
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 546
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 547
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 548
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
NON 549
DCA(K)=DCA(K)+DCA(J)*AIN(J,K)
IF (I*EQ*1) AA(I,L)=AA(I,L)+ONE
980 CONTINUE
990 CONTINUE
CALL D5=SUM (AA,ISMCP,22)
IF (ERR) RETURN
DO 960 I=1,ISMC
II=I+25
DO 910 J=1,ISC
900 GJ(IJ)=GJ(IJ)+AA(I,ISMCP1)
DO 930 K=1,ISC
GJ(K)=0
910 GJ(K)=GJ(K)+CI(J)*AIN(J,K)
DO 920 L=1,ISMCP
920 GJ(K)=GJ(K)-GJ(I)*CDIJ(L,K)
930 CONTINUE
IF (GJ(I)) 740,940,940
940 CONTINUE
C COMPUTE SUPPLEMENTARY FLOW VARIABLES AND DERIVATIVES AT END OF
C STEP (1ND BEGINNING OF NEXT STEP)
C
CALL DERIVS
IF (ERR) RETURN
IFAIL=60
IF (FAILED) GO TO 740
IF (NT*EQ,1) GO TO 960
C FINAL VALIDITY CHECK IN ELECTRONIC NONEQUILIBRIUM MODEL --
C TEST ON CHANGE IN ENERGY TRANSFER TO ELECTRONS
IF (OEOLJ*EQ.0.) GO TO 950
IFAIL=5
DO=AMAX(ABS(OEOLD),DQMAX)
IF (ABS(QDPE-OEOLJ)*GT.DOEM) GO TO 740
DQMAX=AMAX(DQMAX,DOEM)
950 QEOLD=QDPE
C FULLY SUCCESSFUL STEP: SET UP FOR NEXT STEP.
960 SC=SC
TP=CT*CTAP
TEP=CTE*CTAP
IF (IS5*GE.0) GO TO 970
WRITE (6,1140) CX,DELTAX,TP,TEP,CHA,QDPE,1COUNT
IF (1CCOUNT*EQ.0) GO TO 980
WRITE (6,1040) (IF(K),K=1,1COUNT)
970 ICOUNT=0
980 ISTEP=SISTEPS+1
DELTAX=AMAX1(DELTAX,0.7*DXOLD)
DXOLD=DELTAX
GO TO 210
990 WRITE (6,1150)
C ERROR EXIT NO. 10
IERR=10
999 WRITE (6,1070) IERR
CALL DUMP (RNAM
RETURN
NON 551
NON 552
NON 553
NON 554
NON 555
NON 556
NON 557
NON 558
NON 559
NON 560
NON 561
NON 562
NON 563
NON 564
NON 565
NON 566
NON 567
NON 568
NON 569
NON 570
NON 571
NON 572
NON 573
NON 574
NON 575
NON 576
NON 577
NON 578
NON 579
NON 580
NON 581
NON 582
NON 583
NON 584
NON 585
NON 586
NON 587
NON 588
NON 589
NON 590
NON 591
NON 592
NON 593
NON 594
NON 595
NON 596
NON 597
NON 598
NON 599
NON 600
NON 601
NON 602
NON 603
NON 604
NON 605
1010 FORMAT (71H MORE THAN 50 RATE-CONSTANT INCREASES REQUIRED IN PERTURBATION SOLUTION) NON 606
1020 FORMAT (2OH *==CHMIN/DCHMAX=1PE10*3.41H IS TOO SMALL. CONCENTRATION NON 607
OF REACTANT A+16H IN REACTION NO.13,3H IS E10*3.8H MOLE/GMN NON 608
2*36H==RATE CONSTANT FOR REACTION NO.13,25H INCREASED BY A FACTOR NON 609
3CTOR OF E10*3.72H FOR PERTURBATION SOLUTION/) NON 610
1030 FORMAT (36H==RATE CONSTANT FOR REACTION NO.13,27H RESTORED TO NON 611
1 CORRECT VALUE,) NON 612
1040 FORMAT (24H CAUSES OF STEP FAILURES 2014) NON 613
1050 FORMAT (28H =********** CONCENTRATION OF A4,10H FROZEN AT 1PE10*3.9 NON 614
1H MOLE/GMx/11X.20H SUPPRESSED REACTIONS,3413/30X,3013) NON 615
1060 FORMAT (4H=0J!((12.16H)**2 UNDERFLOWED) NON 616
1070 FORMAT (15H=ERRR CR EXIT NO.13,11H FROM NONEQ) NON 617
1080 FORMAT (24H NONEQUILIBRIUM SOLUTION/) NON 618
1090 FORMAT (33H=BETA MATRIX OF INSUFFICIENT RANK) NON 619
1100 FORMAT (18H=DLOGR IS POSITIVE) NON 620
1110 FORMAT (8H=DCHMIN=1PE10*3.10X,7H=DCHMAX=E10*3.10X,5HIMAX=12) NON 621
1120 FORMAT (55H BACKSTEPING OF PERTURBATION SOLUTION TERMINATED AFTER NON 622
1=I3,31H STEPS. DIAGNOSTIC DATA FOLLOW/) NON 623
1130 FORMAT (41HTEMPERATURE GREATER THAN RESERVOIR VALUE) NON 624
1140 FORMAT (3H=0X=1PE12*5.5X,7H=DELTAX=E10*3.5X,2HT=OPF8*2.5X,4HTEP= NON 625
1F8*2.5X,4HCHA=1PE15*7.5X,5H=QDPF=E10*3.5X,7HICOUNT=12) NON 626
1150 FORMAT (6OH NONEQUILIBRIUM INTEGRATION UNABLE TO TAKE A CONVERGENT NON 627
1 STEP) NON 628
1160 FORMAT (33H NONEQUILIBRIUM SOLUTION REQUIRED, 15,18H INTEGRATION ST NON 629
EPS) NON 630
END NON 631

IF (F3-F2) 40,60,80
40 IF (DELT=TESTB) 90,50,50
50 DELT=DELT/2
CT=T1-DELT
GO TO 20
60 IF (DELT=TESTB) 90,70,70
70 T1=T2
F1=F2
DELT=DELT/2
CT=T1-DELT
GO TO 20
80 T1=T2
T2=T3
F1=F2
F2=F3
GO TO 30
90 Z=F1-F2+F3-F2
IF (ABS(Z/F2).GT.00001) GO TO 100
CT=T2
GO TO 110
100 CT=T2+DELT*(F3-F1)/(2*Z)
110 CONTINUE
CALL NEWRAP
IF (ERR) RETURN
SM=FLUX
RTH=RHO
CTMAX=CT
PREST=PRES
DJ 120 I=1, ISS
120 CAPXTH(I)=CAPX(I)
RETURN
END
SUBROUTINE OUT
WRITE (6,10) OUT 1
WRITE (6,20) OUT 2
WRITE (6,30) OUT 3
WRITE (6,40) OUT 4
WRITE (6,50) OUT 5
WRITE (6,60) OUT 6
WRITE (6,70) OUT 7
WRITE (6,80) OUT 8
WRITE (6,90) OUT 9
WRITE (6,100) OUT 10
WRITE (6,110) OUT 11
RETURN

FORMAT (23H1 DEFINITIONS OF SYMBOLS)

1 AREA RATIO / 96H DELREF = EQUILIBRIUM SHOCK STANDOFF DISTANCE ON FOUT
2 LAT-FACED MODEL, DIVIDED BY MODEL NOSE RADIUS / 99H DELREF = EQUILIBRIUM
3 SHOCK STANDOFF DISTANCE ON HEMISPHERICAL MODEL, DIVIDED BY MODEL OUT
4 NOSE RADIUS / 94H DELRH = FROZEN SHOCK STANDOFF DISTANCE ON FOUT
6 SHOCK STANDOFF DISTANCE ON HEMISPHERICAL MODEL, DIVIDED BY MODEL OUT
7 NOSE RADIUS / 73H DELSTR = DISPLACEMENT THICKNESS OF BOUNDARY LAYER
8 ON NOZZLE WALL (INCH)

FORMAT (64H DELSTW = BOUNDARY LAYER DISPLACEMENT THICKNESS ON WEDOUT)
1 GE (INCH) / 35H DIAM = NOZZLE DIAMETER (INCHES) / 56H EPSLE = DENOUT
2 SITY RATIO ACROSS EQUILIBRIUM NORMAL SHOCK / 51H EPSLF = DENSITY ROUT
3ATIO ACROSS FROZEN NORMAL SHOCK / 52H GAMMA = FROZEN SPECIFIC-HEATOUT
4 RATIO IN FREE STREAM / 43H H = ENTHALPY IN FREE STREAM (BTU/LBTOUT)
5A/5B HEIGHT = SECOND TRANSVERSE DIAMETER OF CHANNEL (INCHES) / SECOUT
67H HS = RECOVERY ENTHALPY (BTU/LBTOUT) / 74H HRATE = RATIO OF DISOUT
7SOCIATION ENTHALPY TO (HS-HW) FOR EQUILIBRIUM SHOCK OUT

FORMAT (69H HRATF = RATIO OF DISSOCIATION ENTHALPY TO (HS-HW) FOROUT)
1R FROZEN SHOCK / 39H HS = STAGNATION ENTHALPY (BTU/LBTOUT)
2E = SHOCK-BOUNDARY LAYER INTERACTION PARAMETER PER FOOT FOR EQUOUT
3LIBRIUM SHOCK / 79H K2PF = SHOCK-BOUNDARY LAYER INTERACTION PARAMOUT
4ETER PER FOOT FOR FROZEN SHOCK / 83H LEE = ATOM-MOLECULAR LEWIS NOUT
5UDBER AT STAGNATION CONDITION FOR EQUILIBRIUM SHOCK / 78H LEF = OUT
6AT3H MOLECULAR LEWIS NUMBER AT STAGNATION CONDITION FOR FROZEN SHOUT
7K/37H M = MACH NUMBER IN FREE STREAM / 47H MU = VISCOSITYOUT
9 IN FREE STREAM (LB/FT-SEC) / 73H MUT2E = VISCOSITY AT STAGNATION OUT
3 CONDITION FOR EQUILIBRIUM SHOCK (LB/FT-SEC) / 73H MUT2F = VISCOSITYOUT
44 SY AT STAGNATION CONDITION FOR FROZEN SHOCK (LB/FT-SEC) / 2H MW OUT
5$ = MOLECULAR WEIGHT IN FREE STREAM (GM/MOLE) / 52H MW OUT

FORMAT (83H MUT2E = MOLECULAR WEIGHT AT STAGNATION CONDITION FOROUT)
1 EQUILIBRIUM SHOCK (GM/MOLE) / 79H MUT2F = MOLECULAR WEIGHT AT STADOUT
2GNATION CONDITION FOR FROZEN SHOCK (GM/MOLE) / 53H N = BOUNDAROUT
3Y-LAYER PRESSURE-GRADIENT PARAMETER / 40H P = PRESSURE IN FREEOUT
4 STREAM (ATM) / 71H PRE = PRANDTL NUMBER AT STAGNATION CONDITIONOUT
5 FOR EQUILIBRIUM SHOCK / 65H PRE2 = PRANDTL NUMBER AT STAGNATIONOUT
6 CONDITION FOR FROZEN SHOCK / 68H PRREF = PRANDTL NUMBER AT REFERENOUT
7CE TEMPERATURE IN BOUNDARY LAYER / 58H PT2E = STAGNATION PRESSUREOUT
8 FOR EQUILIBRIUM SHOCK (ATM) / 53H PT2F = STAGNATION PRESSURE FOROUT
9 Frozen shock (atm)/50H Pww = Surface pressure on a wedge modeout 56
$SL (ATM))

60

60 FORMAT (130H QEFF = Stagnation-point heat flux times sort(r) fcout 58
1 H Equilibrium shock on a flat-faced model with an equilibrium boundout 59
220X*40H (BTU/SL FT-SEC)FT**0.5 = stagnation-point heat at shock times sort(r) for equilibrium 60
3F = stagnation-point heat at shock times sort(r) for equilibrium 61
4H shock on a flat-faced model with a frozen boundary layer/20X*40H (BTUOUT 62
5H/SQ FT-SEC)FT**0.5 = FAY-RIDDELL/127H QEFF = Stagnation-point heat flux times sort(r) 63
6H INT FLUX TIMES SORT(r) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICAL 64
7H MODEL WITH AN EQUILIBRIUM BOUNDARY LAYER/20X*40H (BTUOUT 65
8H FT**0.5 = FAY-RIDDELL/127H QEFF = Stagnation-point heat flux times sort(r) 66
9H SORT(r) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICAL MODEL WITH A FOUT 67
$FROZEN BOUNDARY LAYER/20X*41H (BTU/SQ FT-SEC)FT**0.5 = FAY-RIDDELL 68
SHELL = fcout 69

70

70 FORMAT (52H QFCT = net power added to the electron gas (CAL/CU CM)
1H SEC/120H QFCT = stagnation-point heat flux times sort(r) fcout 71
2H Equilibrium shock on a model (BTU/SQ FT-SEC)FT**0.5 = stagnation-point heat times sort(r) for fcout 72
3/125H QFCT = stagnation-point heat flux times sort(r) for frozen shock on model 73
4H shock on a flat-faced model with an equilibrium boundary layer/20OUT 74
5H*41H (BTU/SQ FT-SEC)FT**0.5 = FAY-RIDDELL/119H QFCT = Stagn 75
6G stagnation-point heat flux times sort(r) for frozen shock on a flat-out 76
7H CED model with a frozen boundary layer/20X*41H (BTU/SQ FT-SEC)FT**OUT 77
8H = fcout 78

80

80 FORMAT (129H QHFFFFF = stagnation-point heat flux times sort(r) fcout 79
1R FROZEN SHOCK ON A HEMISPHERICAL MODEL WITH AN EQUILIBRIUM BOUNDOUT 80
2R LAYER/20X*40H (BTU/SQ FT-SEC)FT**0.5 = FAY-RIDDELL/122H QHFFFFF 81
3R = stagnation-point heat flux times sort(r) for frozen shock on model 82
4R a hemispherical model with a frozen boundary layer/20X*40H (BTU/SQOUT 83
5R FT-SEC)FT**0.5 = FAY-RIDDELL/79H QHFFFFF 84
6R Plate 1 ft from the leading edge (BTU/SQ FT-SEC) 85
SHELL = fcout 86

90

90 FORMAT (117H VQFFSR = stagnation-point heat flux times sort(r) fcout 87
1R FROZEN SHOCK ON MODEL (BTU/SQ FT-SEC)FT**0.5 = stagnation-point heat times sort(r) for fcout 88
2H QFCT = radiated power (CAL/CU CM-SEC)/5H QW 89
3R to nozzle wall (BTU/SQ FT-SEC)/64H QW = heat flux to surface 90
4R OF A WEDGE MODEL (BTU/SQ FT-SEC)/44H R = Density in free stou 91
5R STREAM (LB/CU FT)/50H REPFF = REYNOLDS number per foot in free strou 92
6R AM/54H RETH = REYNOLDS number based on momentum thickness/95H OUT 93
7R = critical Reynolds number (based on momentum thickness) fcout 94
8R Boundary layer transition/75H R2E = Density at stagnation 95
9R Nuolution for equilibrium shock (LB/CU FT)/48H S = entropy inout 96
$R STREAM (BTU/LB-DEG R)/56H SIGMA = ELECTRICAL CONDUCTIVITY OUT 97
$IN FREE STREAM (MHO/CM))

100

100 FORMAT (59H SIGT2F = ELECTRICAL CONDUCTIVITY AT STAGNATION CONDITIONS 98
110 FOR EQUILIBRIUM SHOCK (MHO/CM)/64H SIGT2F = ELECTRICAL CONDUCTIVITY 99
2TIVITY AT STAGNATION CONDITION FOR FROZEN SHOCK (MHO/CM)/49H STANTOUT 100
3N = STANTON NUMBER BASED ON TOTAL ENTHALPY/79H ST2F = ENTROPY OUT 101
4AT STAGNATION CONDITION FOR EQUILIBRIUM SHOCK (BTU/LB-DEG R)/74H SOUT 102
5T2F = ENTROPY AT STAGNATION CONDITION FOR FROZEN SHOCK (BTU/LB-OUT 103
6DEG R)/46H T = TEMPERATURE IN FREE STREAM (CEG K))

110

110 FORMAT (79H T2E = TEMPERATURE AT STAGNATION CONDITION FOR EQUIOUT 105
1110 LIBRUM SHOCK (CEG K)/71H T2E = TEMPERATURE AT STAGNATION CONDOUT 106
2TITION FOR FROZEN SHOCK (CEG K)/59H T2E = TEMPERATURE AT STAGNATION 107
3E WALL (LB3/SQ FT)/43H TELEC = ELECTRON TEMPERATURE (DEG K)/69H OUT 108
4THETA = MOMENTUM THICKNESS OF BOUNDARY LAYER ON NOZZLE WALL (INCLUDOUT 109
5H)/43H V = VELOCITY OF FREE STREAM (FT/SEC)/57H # DTH = FOUT 110
QST TRANSVERSE DIMENSION OF CHANNEL (INCHES) / 93H X  = AXIAL DOUT 111
DISTANCE ALONG NOZZLE, MEASURED FROM THROAT AND POSITIVE DOWNSTREAM = AXIAL DISTANCE ALONG NOZZLE, 112
O AVERAGED OVER SEVERAL PRECEDING POINTS DOUT 113
OF THE SOLUTION / 85H XW  = DISTANCE FROM LEADING EDGE OF WEDGE, OUT 114
$M$EASURED ALONG WEDGE SURFACE (INCHES) / 115H YS  = ORDINATE OF SOUT 115
$SHOCK$ FROM LINE PARALLEL TO FREE STREAM FLOW, PASSING THROUGH LEADING 116
EDGE OF WEDGE (INCHES) / 71H ZETA  = NONDIMENSIONAL STREAMWISE OUT 117
COORDINATE IN BLUNT WEDGE ANALYSIS)  OUT 118
END  OUT 119
50  FAREA=2.*RTHCM(1)  
   GO TO 100  
60  IF (NPRFLS*EQ.1) GO TO 80  
   DO 70 I=1,2  
70  DELSTP(I)=(1.-OMDIST(I))*RO  
   FAREA=4.*(RTHCM(1)-DELSTP(I))*(RTHCM(2)-DELSTP(2))  
   GO TO 100  
   IF (JDIV*EQ.0) GO TO 90  
   FAREA=PS*(OMDIST(1)*RTHCM(1))*2  
   GO TO 100  
   FAREA=FAREA/929.03  
   WRITE (6,570)  
   TFLow=FAREA*CF(1)*CF(2)*SM  
   IF (ISK2B*GT.0) GO TO 110  
   WRITE (6,580) FLOW  
   IF (IC*EQ.0) GO TO 120  
   DEL=SAVEC(1)*RHO*TB7/CM  
   GO TO 130  
120  DEL=0.  
130  IF (NPRFLS*EQ.2) GO TO 140  
   WRITE (6,590) FACNAM,DIAM(1)  
   GO TO 150  
140  WRITE (6,600) CHANAM,DIAM(1),DIAM(2),FACNAM  
150  WRITE (6,610) POUT,TOUT,HOUT,SOUT,ROUT,OUT,FLX,TFLow,GAMMA,CM,DEL  
   WRITE (6,620)  
   WRITE (6,630) (HP(I),SAVEC(I),I=1,ISS)  
   RETURN  
160  FVOUT(1)=CF(4)*CX  
   FVOUT(2)=CT*CTAP  
   FVOUT(3)=CF(5)*CH  
   FVOUT(4)=PRES*PRESA  
   FVOUT(5)=CF(1)*RHO  
   FVOUT(7)=CF(2)*SU  
   FVOUT(8)=AMACH  
   FVOUT(9)=SEN  
   FVOUT(11)=AFNX  
   I6=16+1  
   IF (I6*EQ.0) GO TO 180  
   DO 170 I=1,NPRFLS  
170  II=7+10*I  
   FVOUT(II)=CF(6)*DELBL(I)  
   FVOUT(II+1)=CF(6)*THETA(I)  
   FVOUT(II+2)=QWOT(I)  
   FVOUT(II+3)=TAUW(I)  
   FVOUT(II+4)=1.*BHR  
   CONTINUE  
180  VARNAM(6)=DNAM  
   VARNAM(16)=ANAM  
   IF (NPRFLS*EQ.1) GO TO 200  
   CALL G*AR2 (CX,V,Z)  
   IF (ERR) RETURN  
   IF (I6*NE.0) GO TO 190  
   CHDIMS(2)=2.*Y*CF(4)

PREVIOUS PAGE BLANK, NOT FILLED
CALL G*AR
CONTINUE
IF (E2H) RETURN

FVOUT(6)=2.*Y*CF(4)

CALL GMAR (CX,Y)
IF (ERR) RETURN

FVOUT(12)=2.*Y*CF(4)

CALL TRANSP (FVOUT(2),FVOUT(4))
IF (ERR) RETURN

FVOUT(14)=VIS*CF(3)

IF (VISC*EQ*0.) GO TO 230
FVOUT(12)=FVOUT(5)*FVOUT(7)/FVOUT(14)
GO TO 240

FVOUT(12)=0.
FVOUT(15)=SIGMA
GO TO 250

FVOUT(12)=SIGMA
FVOUT(14)=CF(3)*VIS.
FVOUT(PS)=SIGMA
FVOUT(13)=CM
H3=1.8*CHA*CAP/CMA
IF (ISW3B*EQ*0.) GO TO 340
IF (FVOUT(7))*EQ*0.) GO TO 270
STFAC=1./((FVOUT(5)*FVOUT(7))+(H0-HW))

FVOUT(23)=STFAC*FVOUT(19)
FVOUT(24)=FVOUT(12)*FVOUT(18)/12.
FVOUT(25)=20.*EXP(0.224*FVOUT(8))
FVOUT(33)=STFAC*FVOUT(29)
FVOUT(34)=FVOUT(12)*FVOUT(28)/12.

FVOUT(35)=FVOUT(25)

IF (LIMOUT*EQ*0.) GO TO 300
DO 290 I=1,NPRFLS
FVO2(I,1)=SN(1)

FVO2(2,1)=SN(1)

CALL GEOMAR (CX,ARAT,ARAT)
IF (ERR) RETURN

IF (NPRFLS*EQ*2) GO TO 310
FVOUT(16)=ARAT
GO TO 320

FVOUT(6)=ARAT
VARN4(6)=ANAM
GO TO 340

IF (JDIM*EQ*0.) GO TO 330
FVOUT(6)=RTHCM(I)*SORT(ARAT)*CF(4)*2.
GO TO 340

FVOUT(6)=RTHCM(I)*ARAT*CF(4)*2.
INEQ1=INEQ+1
WRITE (6,640) (ASTRKS,I=1,8),(ASOLN(J,J=1,3),I=1,3),I=1,3)OUTI 221
1NSTEPS,STEP(ISCLN),ASTRSK(L=1,4),TYPSLN(INEDEP1,ISDLN) OUTI 222
WRITE (6,650) (VARNAM(I),FVOUT(I),I=1,15) OUTI 223
IF(ISW3B,EQ,0) GO TO 360 OUTI 224
DO 350 L=1,NPRFLS OUTI 225
IF(L=2+1*L) I=11+0 OUTI 226
I=110+L OUTI 227
WRITE (6,660) (VARNAM(I),FVOUT(I),I=1,12) OUTI 228
CONTINUE OUTI 229
360 IF(NPRFLS.LT.10.OR.ISW3B. NE.0) GO TO 370 OUTI 230
WRITE (6,670) CHDIMS(1),CHDIMS(2) OUTI 231
370 IF (LIMOUT.EQ.0) GO TO 380 OUTI 232
WRITE (6,680) ((VARN2(I),J,FV02(I,J),I=1,2),J=1,NPRFLS) OUTI 233
TNOUT(I)=CTAP*CTE OUTI 234
TNOUT(2)=QDPF INOUT(3)=QDPE TNOUT(4)=Q0 OUTI 235
WRITE (6,690) (TNNAME(I),TNOUT(I),I=1,4) OUTI 236
IF (IC.EQ.0) GO TO 400 OUTI 237
IF (IC.EQ.0) GO TO 400 EMF=SAVEC(I) OUTI 238
SAVEC(1)=EMF*RHO*TB7/CM OUTI 239
400 ISSSP1=ISS-11/541 DO 410 K=1,ISSP1 OUTI 240
LIM1=5*(K-1)+1 OUTI 241
LIM2=5*NO(5,K,ISS) OUTI 242
WRITE (6,700) (PSPI(I),SAVEC(I),I=LIM1,LIM2) OUTI 243
IF (IC.EQ.0) GO TO 420 OUTI 244
IF (IC.EQ.0) GO TO 400 SAVEC(1)=EMF OUTI 245
420 IF(ISW5B.GT.0.CR.ISOLN. NE.3) GO TO 440 OUTI 246
WRITE (6,710) (I,PI(I),I=1,ISR) OUTI 247
WRITE (6,520) (I,CHI(I),I=1,ISR) OUTI 248
WRITE (6,530) (I,PCHI(I),I=1,ISR) OUTI 249
DO 430 I=1,ISS OUTI 250
OUTI 251
OUTI 252
OUTI 253
430 DLOGDX(I)=0.GJ(I)/GJ(I) OUTI 254
WRITE (6,550) (PSI(I),DLOGDX(I),I=1,ISS) OUTI 255
IF (ITYPER.NEQ.1) GO TO 500 OUTI 256
ITYPER=2 OUTI 257
IX=0 OUTI 258
440 IF (*NOT*DATAPE) GO TO 500 OUTI 259
IMP=0 OUTI 260
450 IX=IX+1 NRCOUT=NRCOUT+1 OUTI 261
ITYPER=2 OUTI 262
XXX(1)=-FVOUT(1) OUTI 263
XXX(2)=-FVOUT(2) OUTI 264
XXX(3)=-FVOUT(3) OUTI 265
XXX(4)=-FVOUT(4) OUTI 266
XXX(5)=-FVOUT(5) OUTI 267
XXX(6)=-FVOUT(6) OUTI 268
XXX(7)=-FVOUT(7) OUTI 269
XXX(8)=-FVOUT(8) OUTI 270
IF (ISW3N) 460,480,460
460 IBL=17*I*(NBL-1)
XXX(9)=FVOUT(1BL)
IF (NPFRLES=Eq.2) GO TO 470
XXX(8)=FVOUT(16)
GO TO 490
470 XXX(8)=FVOUT(6)
GO TO 490
480 XXX(8)=FVOUT(11)
XXX(9)=0.
490 XXX(10)=FVOUT(12)
WRITE (ITPOUT) ITPER,IX,XXX(I),I=1,10)
RETURN
C
C 1 FORMAT (57X,18HREACTION RATE DATA)
520 FORMAT (3H P,-.11X,8(I4,1PE10.1),6X)
530 FORMAT (4H CHI,10X,8(I4,1PE10.1),6X)
540 FORMAT (6H PICH,1,8(I4,1PE10.1),6X)
550 FORMAT (4H DLG,12X,7(2X,A4,1PE10.1),4X)
560 FORMAT (1X,A3,1H(+,11,3H) =4X,1PE10.3,4X,3(A3,1H(+,11,3H) =4X,E.0.0)
570 FORMAT (1H0)
580 FORMAT (19H GAS FLOW RATE =F10.2,2X,7H LB/SEC)
590 FORMAT (10H NOZZLE = A4,5X,F10.3,2X,21H INCH THROAT DIAMETER)
600 FORMAT (9H CHANNEL ,A4,6X,F10.3,3H BY,F7.3,17H INCH THROAT FOR A4)
610 FORMAT (1H0)
1 1 FORMAT (19H PRESSURE =F10.2,2X,4H ATM/19H TEMPERATURE
1 =F7.3,5X,6H CEG K/19H ENTHALPY =F7.0,5X,7H BTU/LB/
2 ENTRPY =F9.2,3X,13H BTU/LB-DEG R/19H DENSITY =U10.3
3,F12.5,9H LB/CU FT/19H VELOCITY =F7.0,5X,7H FT/SEC/19H MAKU
4 S FLUX =F10.3,2X,13H LB/SQ FT-SEC/19H COMPUTED FLOW =U10.3
5F10.3,2X,7H LB/SEC/19H GAMMA =F10.3/19H MOLECULAR WEIGO
6 H =F9.2,3X.8 GM/MOLE/19H ELECTRON DENSITY =3X,1PE9,2,13H ELECTROU
7720S/C1)
620 FORMAT (/10X,22HSPICES MOLE FRACTIONS/)
630 FORMAT (13X,A4,4X,1PE10.3)
640 FORMAT (/13H,14A,6,6A6/)
650 FORMAT (I3X,A4,1A8,6A6/)
1 H =F4.1PE10.3,4X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
2 H =F9.0,10X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
3 H =F9.0,10X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
4 H =F9.0,10X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
660 FORMAT (13X,A4,1A8,6A6/)
1 H =F4.1PE10.3,4X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
2 H =F9.0,10X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
3 H =F9.0,10X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
4 H =F9.0,10X,A6,2H =F9.0,10X,A6,2H =F9.0,10X,A6,2H
670 FORMAT (9H WIDTH =F11.3,6X,9H HEIGHT =F11.3)
680 FORMAT (1X,A6,2H =F8.0,10X,A6,2H =4X,1PE10.3,4X,A6,2H =4X,10X,A6,2H
1 H =F4.1PE10.3,4X,A6,2H =4X,1PE10.3,4X,A6,2H =4X,10X,A6,2H
212,4X,A6,2H =4X,1PE10.3,4X,A6,2H =4X,1PE10.3,4X,A6,2H
690 FORMAT (4X,41HSPICES MOCLE FRACTIONS IN THE FREE STREAM)
700 FORMAT (5(1X,*.4,3X,1H)=,4X,1PE10.3,3X))
END
FUNCTION POMEG(TL)
COMMON /TNEQ/ TL,TNEQ,TLIST(30),POM(30)
CONTINUE
IF (TNEQ EQ TL(I)) GO TO 30
POMEG=POM(I-1)+POM(I-1)(TE-TLIST(I-1))/TLIST(TLIST(I-1))
RETURN
END
SUBROUTINE PROP
REAL ACOM(30), ELMENT(10)*HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22:24), CDIJ(20:10), CAPX(20)
COMMON A, AFNX, AFNS, AMACH, AR, ARBA, ARBB, BZERD
COMMON 1, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP
COMMON 2, CRR, CSTA, CT, CTP, CTR, CTC, CTMAX, CTMXK
COMMON 3, CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT
COMMON 4, DELT2, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT
COMMON 5, PCTest, PRES, PRESA, PRESB, PRESRH, PRHO, RHAP, RHO, RHOB
COMMON 6, RHOBAR, RHOC, RHOP, RHPL, RHM, ROBAR, ROBARP, SCRP, SDT
COMMON 7, SEN, SHAP, SC, SL, SL64, SM, SU, SU2, SUMG
COMMON 8, TEST, TESTB, PRINT, TSTOP, UP, ZP, ZPA
COMMON BE(64), BET(20), BLBK(31), CAl(64), CAPAHT(20)
COMMON CCPJ(20), CEA(CA(6)), CGMU(20), CHI(64), CHII(20)
COMMON 2, CLNMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJK(20)
COMMON GJ(20), GJER(20), PI(64), PChi(64), GM(20)
COMMON 4, QO(64), SAJ(20), SDCH(64), SENT(20), SHJ(20), SHJA(20)
COMMON 5, SKL(20), SS(20), TR(30), TF(20), TFE(20)
COMMON 6, TF0(20), TFE(20), TF1(20), THEV(20), XMJAT(20), XNU1(64)
COMMON 7, BE(64), BET(20), BLBK(31), CAl(64), CAPAHT(20)
COMMON CCPJ(20), CEA(CA(6)), CGMU(20), CHI(64), CHII(20)
COMMON 2, CLNMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJK(20)
COMMON GJ(20), GJER(20), PI(64), PChi(64), GM(20)
COMMON 4, QO(64), SAJ(20), SDCH(64), SENT(20), SHJ(20), SHJA(20)
COMMON 5, SKL(20), SS(20), TR(30), TF(20), TFE(20)
COMMON 6, TF0(20), TFE(20), TF1(20), THEV(20), XMJAT(20), XNU1(64)
COMMON IP, IRUN, ISC, ISCP
COMMON 1, ISMC, ISMCN, ISR, ISS, ISSNR, ISEP1, ISEP2, ISEP3, ISEP4
COMMON 2, ISW1A, ISW1B, ISW2A, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A
COMMON 3, ISW6B, ISW6A, IUP, IZERO, JIK, LC, M, NFIT
COMMON NIT, NNN, NNS, NO1, NTEST
COMMON IGJ(20), IGM(20), ITB(5), KUR(64,20), LPIJ(20,10)
COMMON ACOM, ELMENT, HP
COMMON DIMENSION AAA(22:24), BTA(64,20), CAP0(31), CCI(20), DGAJ(20), GJ(20), SBP
COMMON 1, SDGJ(20), SHJAP(20), THEVP(20)
COMMON EQUIVALENCE (AA(1,1), AAA(1,1), (BETA(1,1), BTA(1,1)), (DLBK(1,1), CAP0(1)), (CAPAHT(1), CCI(1)), (GJ(1), DGAJ(1)))
COMMON (CAPX(1), GJ(1)), (SAJ(1), SB(1)), (SS(1), SDGJ(1))
COMMON (SHJA(1), SHJAP(1)), (THEV(1), THEVP(1))
COMMON DATA RNAME /AMFRC/
IF (ERR) RETURN
NSTEPS=0
IF (.NOT.MODLPT) GO TO 30
CALL MODEL MFLDLPT=eFALSEa
30 IF (INEO.NE.0) GO TO 40
CT=CT-PCT
CTE=CT
RHO=RHO-PRHO
40 RETURN
C
END
SUBROUTINE PUTQTN (X)
SUBROUTINE PUTQTN SUPERVISES THE COMPUTATION OF COLLISION CROSS SECTIONS FOR THE TRANSPORT CALCULATIONS.

LOGICAL WROTE
DIMENSION OM(3)
COMMON /TRANS1/ TQO(3,20,20),ZM2(20)
COMMON /TRANS7/ V(400),KO(100),NO(100),IQ(400),JQ(400),NKO
COMMON /TRANS8/ NV(15),N
COMMON /RDTR/ ISW8B
DATA IDBG /-1/
IDBG=IDBG+1

WROTE=FALSE.

Initialize Q array to zero.
DO 10 I=1,N
DO 10 J=1,N
DO 10 K=1,3
Q(K,1,J)=0.
CALL QCoul (OM,X,V)
Q(1,2,1)=OM(1)

Compute cross section values and accumulate in Q array.
M=2
LQ1=1
DO 20 L=1,NKO
K=KQ(L)
LQ2=KQ(L)
GO TO (20,30,40,50,60,70,80,90,100,110,120,130,140,150,160), K
20 CONTINUE
GO TO 190
30 OM(1)=V(M)*Q(1,2,1)
OM(3)=OM(1)*1.5625
OM(2)=V(M+1)*OM(1)
GO TO 190
40 CALL QEXP (OM,V(M))
GO TO 190
50 CALL QZj (OM,V(M))
IF (V(M+3)) 190,190,170
60 CALL QZa (OM,V(M))
GO TO 190
70 CALL Qzj (OM,V(M))
GO TO 190
80 CONTINUE
17 CALL QATTP (OM,V(M))
GO TO 190
90 CALL QLJ (OM,V(M))
GO TO 190
100 CALL QSAME (OM,V(M))
GO TO 190
110 CALL QMX (LQ1,LQ2)
GO TO 210
120 CALL Q11 (LQ1,LQ2,V(M))
GO TO 210
130 CALL Q12 (OM,V(M))
GO TO 190
140 CALL Q13 (LQ1,LQ2,V(M))
GO TO 210
CALL Q14 (LQ1,LQ2,V(M))
150 CONTINUE
GO TO 210
170 DO 180 L0=LQ1,LQ2
I=IQ(L0)
J=IQ(L0)
Q(I+1,J)=QM(1)
180 Q(3+1,J)=QM(3)
GO TO 210
190 DO 200 L0=LQ1,LQ2
I=IQ(L0)
J=IQ(L0)
DO 220 KK=1,3
200 Q(KK+I,J)=Q(KK+I,J)+QM(KK)
210 M=M+NV(K)
C DEBUG OUTPUT
IF (ISWBB.EQ.0) GO TO 240
IF (IDBG.EQ.0) GO TO 220
IF (IDBG.NE.ISWBB) GO TO 240
220 WROTE=TRUE
WRITE (6,250) L,K,NO(L),M,LQ1
DO 230 I=1,N
230 WRITE (6,260) (Q(I,J),J=1,1)
240 LQ1=LQ2+1
IF (WROTE) IDBG=0
RETURN
C 250 FORMAT (1HC/10/I5/)
260 FORMAT (1P10E12.4)
END
SUBROUTINE OC0UL(OM1*1X1*)
SUBROUTINE QC0UL COMPUTES THE FACTOR 0.8*QC IN THE EFFECTIVE
COULOMB CROSS SECTION.

COMMON/TRANSL/T,Q(3,20),ZM2(20)
RETURN
END
SUBROUTINE QEX (OM*,V*)
SUBROUTINE QEX COMPUTES CROSS SECTIONS FOR A RESONANT EXCHANGE PROCESS.
DIMENSION OM(3),V(3)
COMMON /TRANS1/ T,0(3,20,20),ZM2(20)
A=V(1)
B=V(2)
ZM=V(3)
Y=A-217147*B*ALOG(T/ZM)
OM(1)=2*(Y-4.461*B)**2+0.3724*B**2
OM(3)=2*(Y-4.317*B)**2-0.6754*B**2
OM(2)=0.
RETURN
END
SUBROUTINE QEXP (OM, V)

SUBROUTINE QEXP COMPUTES CROSS SECTIONS FOR AN EXPONENTIAL
INTERACTION POTENTIAL.
DIMENSION OM(3), V(3)
COMMON /TRANS1/ T, Q(3, 20, 20), ZM2(20)
A=V(1)
RHO=V(2)
N1=V(3)
ALPHA=ALOG(A/T)
CALL QINTRP (ALPHA, OM, N1, 50)
OM(1)=12.5664*(ALPHA*RHO)**2*OM(1)
OM(2)=OM(2)*OM(1)
OM(3)=OM(3)*OM(1)
RETURN
END
SUBROUTINE QINTRP (A,B,N1,N)
C SUBROUTINE QINTRP INTERPOLATES TABULAR DATA FOR THE CROSS SECTION
C COMPUTATIONS
COMMON /ROTR/ ISWBB
DIMENSION B(3)
COMMON /TRANS/ TL(1000),OM(1000,3)
I=N1
IF (TL(I)-A) 10,40,30
10 N2=N1+N-2
DO 20 I=N1,N2
IF (TL(I+1)-A) 20,40,40
20 CONTINUE
I=N2
IF (ISWBB.EQ.0) GO TO 40
WRITE (6,60) I,A,TL(I),TL(I+1)
40 DO 50 J=1,3
50 B(J)=OM(I+1,J)+(A-TL(I+1))*(OM(I,J)-OM(I+1,J))/(TL(I)-TL(I+1))
RETURN
C FORMAT (25H0 EXTRAPOLATED VALUESI6,1P4E14.4/1H )
END
SUBROUTINE OLJ (OM,V)
SUBROUTINE OLJ COMPUTES CROSS SECTIONS FOR THE LENNARD-JONES 6-12
DIMENSION OM(3), V(3)
COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
EPSLN=V(1)
SIGMA=V(2)
N1=V(3)
TSTAR=1/EPSLN
CALL QINTRP (TSTAR,OM,N1,37)
OM(1)=3.1416*SIGMA**2*OM(1)
OM(2)=OM(2)*OM(1)
OM(3)=OM(3)*CM(1)
RETURN
END
SUBROUTINE OMMIX (L1, L02)

DIMENSION SORT20, TQ(3, 20), ZM(20)

COMMON TRANS1, TRANS2, V(4), KG(100), NQ(100), IG(400), JG(400), NKG

GO TO 10

10 IF (Q1 = 0) GO TO 30

GO TO 70

10

30

GO TO 70

70

RETURN

END
SUBROUTINE GREPP (OM, VV)
SUBROUTINE GREPP COMPUTES CROSS SECTIONS FOR AN INVERSE POWER LAW
INTERACTION POTENTIAL.
DIMENSION OM(3), VV(3)
COMMON /TRANS1/ T1(3),20,20), ZM2(20)
COMMON /TRANS2/ TL(1000), OMEGA1(1000), ASTAR(1000), BSTAR(1000)
ITL=VV(1)
OM(1)=OMEGA1(ITL)*T1(-2, VV(2))
OM(2)=ASTAR(ITL)*OM(1)
OM(3)=BSTAR(ITL)*OM(1)
RETURN
END
SUBROUTINE OSAME (CM, VV)
SUBROUTINE OSAME SETS THE CROSS SECTION FOR A GIVEN SPECIES PAIR OSA 1
DIMENSION CM(3), VV(3)

COMMON / TRANSF / T, Q(3, 20, 20), ZM2(20)
I=VV(1)
J=VV(2)
CM(1)=CM(1)+C*Q(1, I, J)
CM(2)=CM(2)+C*Q(2, I, J)*VV(4)
CM(3)=CM(3)+C*Q(3, I, J)*VV(5)
RETURN
END
SUBROUTINE QTAB (OM, V)
C USE TABULAR DATA FOR CROSS SECTIONS.
DIMENSION CM(3), V(3)
COMMON /TRANS$/ T, 9(3,20,20), ZM2(20)
A = V(1)
N1 = V(2)
NL = V(3)
CALL QINTRP (T, CM, N1, NL)
OM(1) = A * CM(1)
OM(2) = A * CM(2)
OM(3) = OM(3) * OM(1)
RETURN
END
SUBROUTINE Q11 (LQ1,LQ2,VI)
SUBROUTINE Q11 MULTIPLIES CROSS SECTIONS BY A RAMP FUNCTION OF
TEMPERATURE.
DIMENSION VV(3),VI(2)
COMMON /TRANS1/ T,O(3, 20, 20), ZM2(20)
DATA VV/3*1./
T0=VI(1)
TI=VI(2)
VV(1)=AMAX1(O,(T-T0)/(T1-T0))
IF (VV(1).LT.1.) CALL Q14 (LQ1, LQ2, VV)
RETURN
END
SUBROUTINE Q12 (OM,VV)

SUBROUTINE Q12 COMPUTES CROSS SECTIONS FROM THE GENERALIZED MIXING RULE (EQ. 27).

DIMENSION OM(3),VV(4)

COMMON /TRANS1/ T(3,20*20),ZM2(20)

I=VV(1)
J=VV(2)
K=VV(3)
L=VV(4)

DO 10 M=1,3

OM(M)=.25*(SORT(Q(M,I,J))+SORT(Q(M,K,L)))**2

10 CONTINUE

RETURN

END

SUBROUTINE 013 (L01,L02,VV)
SUBROUTINE 013 SETS TWO CROSS SECTIONS EQUAL FOR A SPECIES PAIR.
DIMENSION VV(3)
COMMON /TRANS1/ T,Q(3*20,20),ZM2(20)
COMMON /TRANS7/ V(400),KQ(100),NG(100),IQ(400),JQ(400),NKQ
K1=VV(1)
K2=VV(2)
C=VV(3)
DO 10 L0=L01,L02
I=IQ(L0)
J=JQ(L0)
Q(K1,I,J)=C*Q(K2,I,J)
RETURN
END
SUBROUTINE Q14 (LQ1,LQ2,VV)
C SUBROUTINE Q14 MULTIPLIES CROSS SECTIONS BY A CONSTANT.
DIMENSION VV(3)
COMMON /TRANS1/ T,Q(3,20,20),ZM2(20)
COMMON /TRANS7/ V(400),KQ(100),NQ(100),IQ(400),JN(400),NKQ
C2=VV(1)*VV(2)
C3=VV(1)*VV(3)
DO 10 L0=LQ1+LQ2
I=IQ(L0)
J=JQ(L0)
Q1(I,J)=VV(1)*Q(1,I,J)
Q2(I,J)=C2*Q(2,I,J)
Q3(I,J)=C3*Q(3,I,J)
10 RETURN
END
SUBROUTINE RADIUS (ITYPE, X, R, AG, AGJ, L)
DIMENSION YZ(2), NPR(2)
COMMON /BL/ DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
SW, R0, JOIM, IP0INT
COMMON /BLRAD/ Y0ZO
COMMON /AREA/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
NPRFLS, NSFCTU(2), ISHAPE(12,2), NPROFL(2), NPRFLS, NBL
DATA NPR /2,1/
GO TO (10, 10, 10, 44), ITYPE
CALL GMAR (X, R)
GO TO (20, 30, 30), ITYPE
AG = RATIO
AGJ = 1
GO TO 50
AGJ = AG
GO TO 50
CALL GMAR2 (X, YZ(1), YZ(2))
AG = YZ(1) * YZ(2) / Y0ZO
R = YZ(L)
M = NPR(L)
AGJ = YZ(L) ** 2
RETURN
END
SUBROUTINE READ
LOGICAL SUPG0, READXS, DATAPE, READG, AAMS, SETIGS
LOGICAL WEDGEM, AXSYM, AXIMOD, NOTRAN
REAL SHAP0(4, 3), SPNAME(30)
REAL ACOM(10), ELMENT(10), HP(20)
DOUBLE PRECISION AA, AA, CAPX, GJ, CDIJ
COMMON AA(22, 24), CDIJ(20, 10), CAPX(20)
COMMON 8 AFNTS, AFNX, AMACH, AR, ARB, BZERO
1 CARB, CH, CLM, C, CM, CRA, CRP
2 CRR, CRS, CSTA, CT, CTAP, CTB, TCT, CTMAX, CTMXX
3 CTP, CTPL, CT, CX, CB, CMAX, DATED, DTEST, DELT1
4 DELTA, DELTA, DLOGA, DLOGR, OT, ENT, FLUX, HDELX, PCT
5 PCTEST, PCTEST, PREST, RPASP, RHO, RHOB
6 RD驱, RHOC, RHOP, RHPL, RHTH, RROBAR, ROBARP, SCBG, SOT
7 SEN, SHPG, SC, SL, SL64, SM, SU, SU2, SUMG
8 TEST, TEST, TPRINT, TSTOP, UP, ZP, ZPA
COMMON BET(64), DET(20), BLK(31), CAI(64), CAIDXH(20)
1 CCPJ(20), CEACT(64), CGI(20), CMU(20), CHI(64), CHII(20)
2 C1NMC(64), CLPNI(64), CMW(20), ETAI(64), ETAJ(20)
3 GJUB(20), PERTGJ(20), PGJ(20), PI(64), PICHI(64), GM(20)
4 QJ(64), SAIJ(20), SDCHI(64), SENT(20), SHJ(20), SHJAI(20)
5 SKIL(20), SSS(20), TB(30), TFA(20), TFB(20), TFC(20)
6 TFD(20), TFE(20), TFK(20), THV(20), XMJAT(20), XNUT(64)
COMMON BETA(64, 20), EL(10, 20), GEL(10, 20)
1 XNU1J(64, 20), XNU1JP(64, 20)
COMMON IC, IM, INEO, INE0, IP, IRUN, ISC, ISCP1
1 ISMC, ISMCN, ISRA, ISSP1, ISSP2, ISSP3, ISSP4
2 IS8A, ISW1A, ISW2A, ISW3A, ISW4A, ISW5A
3 IS8B, IS6A, ISW6A, IPD, IZFR, JJK, LC, ML, NFIT
4 NIT, NNN, NNS, NOS, NOT, NOTEST
COMMON IGJ(20), IGX(20), ITS(5), KUR(64, 20), LPX(20, 10)
COMMON ACOM, AMENT, HP
COMMON /AREA/ ATPL(11, 2), PARAM(3, 12, 2), RTHCM(2), NSECT(2)
1 NSECTU(2), ISHAPF(1, 2), NPROFL(2), NFPFLS, NBL
COMMON /AVG/ WSPE
COMMON /BDEL/ BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2)
1 SROJ, IM, IPINT
COMMON /COLDSP/ CGW1, GJ(10), IJSCS(10), NCS
COMMON /ELEM/ EPI(2), EEPI(2), EPP(2), EEPP(2), EEP5(2), EEP6(2)
1 EEP7(2), EEP8(2), EEP9(2), EEP10(2)
COMMON /FLM/ EPI(2), EEPI(2), EPP(2), EEPP(2), EEP5(2), EEP6(2), EEP7(2)
1 EEP8(2), EEP9(2), EEP10(2)
COMMON /LN/ ISADT, ISMOL, JATOM, JMOL
COMMON /MASSFL/ SMASS, CTMX(1001), TSTOP, IS(20)
COMMON /MIX/ G(124), GP1(124), GP2(124), GP3(124), GP4(124), GP5(124)
1 GP6(124)
COMMON /MOPPAR/ XMPP, DPP, FSTAG, CATFAC, TMOEFO, XMODP1, XMODP2
1 TPLATE, KDIM
COMMON /MOPT/ TSDIAM(20), TSCR(20), NTS, NBL
COMMON /NWM/ FACP, NMODP
COMMON /NOZ/ ZPI(64), ZPZ(64), ZP2(64), ZP4(64), ZP5(64), ZP6(64)
1 ZP7(64), ZP8(64), ZP9(64), ZP10(64), ZP11(64), ZP12(64), ZP13(64)
2 ZP14(64), ZP15(64), ZP16(64), ZP17(64), ZP18(64), ZP19(64)
3 ZP20(64)
<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>161(5)</td>
<td>I62(5)</td>
<td>I63(5)</td>
<td>I64(5)</td>
<td>I65(5)</td>
<td>REA 111</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I66(5)</td>
<td>I67(5)</td>
<td>I68(5)</td>
<td>I69(5)</td>
<td>I70(5)</td>
<td>REA 112</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I71(5)</td>
<td>I72(5)</td>
<td>I73(5)</td>
<td>I74(5)</td>
<td>I75(5)</td>
<td>REA 113</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I76(5)</td>
<td>I77(5)</td>
<td>I78(5)</td>
<td>I79(5)</td>
<td>I80(5)</td>
<td>REA 114</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I81(5)</td>
<td>I82(5)</td>
<td>I83(5)</td>
<td>I84(5)</td>
<td>I85(5)</td>
<td>REA 115</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I86(5)</td>
<td>I87(5)</td>
<td>I88(5)</td>
<td>I89(5)</td>
<td>I90(5)</td>
<td>REA 116</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I91(5)</td>
<td>I92(5)</td>
<td>I93(5)</td>
<td>I94(5)</td>
<td>I95(5)</td>
<td>REA 117</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I96(5)</td>
<td>I97(5)</td>
<td>I98(5)</td>
<td>I99(5)</td>
<td>1100(5)</td>
<td>REA 118</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J6(5)</td>
<td>J7(5)</td>
<td>J8(5)</td>
<td>J9(5)</td>
<td>J10(5)</td>
<td>REA 120</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J56(5)</td>
<td>J57(5)</td>
<td>J58(5)</td>
<td>J59(5)</td>
<td>J60(5)</td>
<td>REA 130</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J61(5)</td>
<td>J62(5)</td>
<td>J63(5)</td>
<td>J64(5)</td>
<td>J65(5)</td>
<td>REA 131</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J71(5)</td>
<td>J72(5)</td>
<td>J73(5)</td>
<td>J74(5)</td>
<td>J75(5)</td>
<td>REA 133</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J76(5)</td>
<td>J77(5)</td>
<td>J78(5)</td>
<td>J79(5)</td>
<td>J80(5)</td>
<td>REA 134</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J91(5)</td>
<td>J92(5)</td>
<td>J93(5)</td>
<td>J94(5)</td>
<td>J95(5)</td>
<td>REA 137</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>J96(5)</td>
<td>J97(5)</td>
<td>J98(5)</td>
<td>J99(5)</td>
<td>J100(5)</td>
<td>REA 138</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMMON /TRAN1**

**COMMON /TRAN2**

**COMMON /TRAN3**

**COMMON /TRAN4**

**COMMON /TRANV**

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1(5)</td>
<td>V2(5)</td>
<td>V3(5)</td>
<td>V4(5)</td>
<td>V5(5)</td>
<td>REA 140</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V6(5)</td>
<td>V7(5)</td>
<td>V8(5)</td>
<td>V9(5)</td>
<td>V10(5)</td>
<td>REA 141</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V16(5)</td>
<td>V17(5)</td>
<td>V18(5)</td>
<td>V19(5)</td>
<td>V20(5)</td>
<td>REA 143</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V21(5)</td>
<td>V22(5)</td>
<td>V23(5)</td>
<td>V24(5)</td>
<td>V25(5)</td>
<td>REA 144</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V26(5)</td>
<td>V27(5)</td>
<td>V28(5)</td>
<td>V29(5)</td>
<td>V30(5)</td>
<td>REA 145</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V31(5)</td>
<td>V32(5)</td>
<td>V33(5)</td>
<td>V34(5)</td>
<td>V35(5)</td>
<td>REA 146</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V41(5)</td>
<td>V42(5)</td>
<td>V43(5)</td>
<td>V44(5)</td>
<td>V45(5)</td>
<td>REA 148</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V46(5)</td>
<td>V47(5)</td>
<td>V48(5)</td>
<td>V49(5)</td>
<td>V50(5)</td>
<td>REA 149</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V51(5)</td>
<td>V52(5)</td>
<td>V53(5)</td>
<td>V54(5)</td>
<td>V55(5)</td>
<td>REA 150</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V56(5)</td>
<td>V57(5)</td>
<td>V58(5)</td>
<td>V59(5)</td>
<td>V60(5)</td>
<td>REA 151</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V61(5)</td>
<td>V62(5)</td>
<td>V63(5)</td>
<td>V64(5)</td>
<td>V65(5)</td>
<td>REA 152</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V66(5)</td>
<td>V67(5)</td>
<td>V68(5)</td>
<td>V69(5)</td>
<td>V70(5)</td>
<td>REA 153</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V71(5)</td>
<td>V72(5)</td>
<td>V73(5)</td>
<td>V74(5)</td>
<td>V75(5)</td>
<td>REA 154</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V76(5)</td>
<td>V77(5)</td>
<td>V78(5)</td>
<td>V79(5)</td>
<td>V80(5)</td>
<td>REA 155</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V81(5)</td>
<td>V82(5)</td>
<td>V83(5)</td>
<td>V84(5)</td>
<td>V85(5)</td>
<td>REA 156</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V86(5)</td>
<td>V87(5)</td>
<td>V88(5)</td>
<td>V89(5)</td>
<td>V90(5)</td>
<td>REA 157</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V91(5)</td>
<td>V92(5)</td>
<td>V93(5)</td>
<td>V94(5)</td>
<td>V95(5)</td>
<td>REA 158</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>V96(5)</td>
<td>V97(5)</td>
<td>V98(5)</td>
<td>V99(5)</td>
<td>V100(5)</td>
<td>REA 159</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION**

AAA(22,24), BTA(64,20), CAP(31), CCI(20), DGJ(20), GJ(20), SBREA 111

**J(20), SDGJ(20), SHJAPI(20), THEVP(20)**

**DIMENSION**

ASYM(0)

DIMENSION CP(5,5)

DIMENSION DUM(A)
DIMENSION EEPRP(2,10)  
DIMENSION ELNAME(3),NATOM(3),EPRP(12,4,6),PRP(29,92),GPRP(124,6).  
DIMENSION NSECTC(2), WORD2(2)  
DIMENSION SY1(3), SY2(3)  
DIMENSION NSECTC(2)  
DIMENSION WCRC(3,2)  
DIMENSION kORD2(2)  
EQUIVALENCE (AA(1,1),AAA(1,1),(BETA(1,1),STA(1,1))  
1  
(ELEK(1),CAPQ(1))  
2  
(CAPX(1),GJ(1))  
3  
(SHJA(1),SHJAP(1))  
&  
EQUIVALENCE (CP1(1),CP1(1))  
EQUIVALENCE (DUM(1),SUMGH)  
EQUIVALENCE (EPRP(1,1),EPRP(1,1))  
EQUIVALENCE (GPI(1),GPRP(1,1))  
EQUIVALENCE (NOZZLE,NPROFL(1,1))  
EQUIVALENCE (NP1(1),PRP(1,1))  
EQUIVALENCE (SP1(1),SPRP(1,1))  
EQUIVALENCE (TNEP(1,1),TN1(1,1))  
EQUIVALENCE (ZP1(1),ZPRP(1,1))  
DATA ASYM /1H,2HPE,2HLI,2HBE,1H8,1HC,1HN,1HF,2HNE,2HNA,2HMG/  
1  
2HAL,2HSI,1H0,1HS,2HCL,2HAR,1HK,2HCA,2HSC,2HTI,1HV,2HCR,2HMN/  
2  
2HFE,2HCO,2HNT,2HCU,2HN2,2HGA,2HGE,2HAS,2HRL,2HBR,2HMR,2HMN/  
2  
2HSL,1H2,2HZR,2HNE,2HM,2HTC,2HRU,2HRO,2HRD,2HAG,2HCD,2HIN/  
2  
2HSN,2HSE,2HTE,1H1,1H2,2HCS,2HBA,2HLA,2HCE,2HNP,2HND,2HPR/  
2  
2HSM,2HEU,2HGD,2HTB,2HYD,2HHC,2HER,2HTM,2HYB,2HLU,2HFL,2HTA/  
2  
1Hw,2HPE,2PCS,HIR,2PRT,2HUA,2HUG/  
DATA IUPOI /1/  
DATA ITS2P /3*H+/ONE/1/  
DATA SHAPD /3*STF,*3*IHT,*3*LIN,*,3*CIRC,*3*LE B,*,3*OTTO,*,3*M*/  
1  
*CIRC,*,3*LE T,*,3*OP,*,3*/  
DATA SPNAME /*SPI,*,3*SPI2,*,3*SPI3,*,3*SPI4,*,3*SPI5,*,3*SPI6,*,3*SPI7,*,3*SPI8,*,3*SPI9,*,3*SPI10,*,3*SPI11,*,3*SPI12,*,3*SPI13,*,3*SPI14,*,3*SPI15,*,3*SPI16,*,3*SPI17,*,3*SPI18,*,3*SPI19,*,3*SPI20,*,3*SPI21,*,3*SPI22,*,3*SPI23,*,3*SPI24,*,3*SPI25,*,3*SPI26,*,3*SPI27,*,3*SPI28,*,3*SPI29,*,3*SPI30/*  
DATA SY1,SY2 /3*H+/1,3*H+/1  
DATA WOKO /3*HTWO-DI,3*MNENSI0,3*MNAL,6H,AXIS,6HYMMETR,3*HIC/  
DATA WORD? /3*HIC,4*DIAM/  
NAMELIST /INPUT/  
C  
*******GROUP 1 - GENERAL CONTROL VARIABLES  
1  
IS1A,IS2A,IS3A,IS4A,IS5A,IS6A,IS8,IS10,IS12,ISW3B,TWALL,NOTRAN/  
1  
TSTOPI,CXMAXI,READGS,READXS/  
C  
*******GROUP 2 - OUTPUT CONTROLS  
2  
IS5H,IS7B,TPRT1,DATAF,PRECOP,IRUN/  
C  
*******GROUP 3 - RESERVOIR CONDITIONS  
3  
ISW2B,PRESFL,LOWX,CTAPI,HSTAG,HTFILTER/  
C  
*******GROUP 4 - GEOMETRY  
4  
NJZLLE,iCHN,JDIM,NPFLS,NPROFL,NBL,DIAM,NSECTC,ISHAPE/  
4  
PARAM1,ATP1,XXEROI/  
C  
*******GROUP 5 - GAS MODEL  
5  
IGAS,AAMS,ACS,JC5,OPJ,ISCI,ISIS,ISRI,ICI,IE,IS,IR,ISATOM/  
5  
ISMDL,CTMXI,XXEROI,INEOVI/
190 IF (ICHAN.GT.0) NBL=CP(4,ICHAN)
200 IF (ISW=96*0) TSW=1
LIMCUT=ISW*8*ISW*30
INEQ=INEQ*1
CTAP=CTAPI
PRES=PRESAI
CTMXX=CTMXXI
CMAX=CXXMAXI*2.5
MBL=1
IF (NPRLS.EQ.1) GO TO 210
IF (MBL.LT.1) MBL=2
210 NTS=0
DO 220 I=1,20
IF (TSDIAM(I).LE.DIAM(MBL)) TSDIAM(I)=1.E30
IF (TSDIAM(I).GT.9.E19) GO TO 230
NTS=NTS+1
IF (NPRLS.EQ.2) GO TO 220
TSAR(NTS)=(TSDIAM(I)/DIAM(1)+JDI+1)
GO TO 230
220 TSAR(NTS)=1.27*TSDIAM(I)
230 CONTINUE
IF (NANGLE.EQ.0.OR.NRACLE.EQ.0) GO TO 240
IF (NX*E=0.0.AND.WX(I).GT.1.E19) GO TO 240
IF (NPRLS.EQ.2) GO TO 240
#EDGFM=#TRUE.
GO TO 250
240 #EDGEM=#FALSE.
250 IF (NOZ/LE.GT.0) GO TO 260
ZZERO=2.54*XZERO
GO TO 270
260 ZZERO=ZPPP(2,NOZLE)
C IN THE CASE OF A CHANNEL, THE CODE USES THE ZZERO FOR THE FIRST
C PROFILE
270 SL=1.
ZZERO=ZZERO1
TSTOP=TSTOP1
TPRINT=TPRNT1
DELTAX=DETXI
DELT1=DELT1I
DO 320 J=1,NPRLS
NPRLS=NPRFL(J)
IF (NPRLS.EQ.0) GO TO 300
RTHCM(J)=ZPRP(1,NPRLF)
NSECUT(J)=ZPRP(3,NPRLF)
NSECJO(J)=ZPRP(4,NPRLF)
NSECJ(NSECUT(J)+NSECJO(J))=ZPRP(64,NPRLF)
NSECJ=NSECJ(J)
DO 290 I=1,NSECJ
ISHPAE(I,J)=ZPRP(I+4,NPRLF)
IF (I.LT.NSFCTJ) ATPI(I,J)=ZPRP(I+16,NPRLF)
DO 280 L=1,3
LL=3*I+L+24
280 CONTINUE
290 CONTINUE

30 NSECTU(J)=NSECTS(1,J)
NSECTD(J)=NSECTS(2,J)
NSECT(J)=NSECTU(J)+NSECTD(J)
IF (J,GE,1) FACNAM=ACOM(1)
RTHCM(J)=1.27*DIAM(J)
NSECTJ=NSECT(J)
DO 310 I=1,NSECTJ
DO 310 L=1,3

310 IF (ICHAN,GE,0) GO TO 330
CHANAM=CP(3,ICHAN)
FACNAM=CP(5,ICHAN)
GO TO 340

320 CONTINUE
IF (ICHAN,LT,0) GO TO 330
CHANAM=CP(1,ICHAN)
FACNAM=CP(4,ICHAN)
GO TO 340

330 CONTINUE
IF (NPRLS,GE,1) GO TO 360
IF (ICHAN,NE,0) GO TO 350
WRITE (6,1030) CHANAM,FACNAM,DIAM(1),DIAM(2)
GO TO 390

340 CONTINUE
IF (NPRLS,LT,1) GO TO 360
WRITE (6,1050) (WORD(I,JDP1),I=1,3),DIAM(1),DIAM(2)
GO TO 390

350 WRITE (6,1040) CHANAM,ICHAN,FACNAM,DIAM(1),DIAM(2)
GO TO 390

360 JDP1=JDIM+1
WRITE (6,1100) (WORD(I,JDP1),I=1,3),DIAM(1),FACNAM
GO TO 390

370 WRITE (6,1100) (WORD(I,JDP1),I=1,3),NOZZLE,DIAM(1),DIAM(2),FACNAM

INAM

380 IF (NOZZLE,LE,NOZZLE.AND.NOZZLE,NE,0) GO TO 440
GO TO 400

390 IF (ICHAN,LE,ICHAN.AND.ICHAN,NE,0) GO TO 440
DO 430 J=1,NPRLS
IF (NPRLS,LT,1) GO TO 410
WRITE (6,1070) J

410 WRITE (6,1090) RTHCM(J),XZERO,NSECT(J),NSECTU(J)
WRITE (6,1100) K,ISH*(SHAPD(I,ISH),I=1,4),ATP(K,J),PARAM(L,K,J)

1L=1,3
NSECTJ=NSECT(J)
K=NSECTJ
ISH=ISHAPE(NSECTJ,J)
WRITE (6,1110) NSECTJ,ISH*(SHAPD(I,ISH),I=1,4),PARAM(L,K,J),L=1,3

430 CONTINUE

440 CONTINUE
IF (NEEL,LE,0) GO TO 460
DO 450 I=1,NEEL
IZ=EEPRP(1,I)+0.1
II=EEPRP(I)
EPRI(1,II)=ASYM(IZ)
EPRI(2,II)=EEPRP(2,II)

450 CONTINUE
**IGS=IGAS**

**IF (IGAS=0.0) GO TO 470**

**WRITE (6,1) 120**

**ISC=ISC**

**ISS=ISS**

**ISR=ISR**

**IC=IC**

**GO TO 520**

**IGAS=ABS(IGAS)**

**WRITE (6,1130) 'GAS=GPRP(1,IGAS)**

**SET UP DESCRIPT ON OF GAS IN TERMS OF ELEMENTS, SPECIES, AND REACTIONS**

**ISC=GPRP(2,IGAS)**

**ISS=GPRP(3,IGAS)**

**ISR=GPRP(4,IGAS)**

**IC=GPRP(5,IGAS)**

**GO TO 480**

**IE(I)=GPRP(I+S,IGAS)**

**DO 490 I=1,NCS**

**IF (IGS.LE.0) GO TO 490**

**GPJ(I)=GPRP(I+S,IGAS)**

**DO 500 I=1,NCS**

**IS(I)=GPRP(I+25,IGAS)**

**DO 510 I=1,ISR**

**IR(I)=GPRP(I+45,IGAS)**

**IATOM=GPRP(I+121,IGAS)**

**IATOM=GPRP(I+122,IGAS)**

**INT=GPRP(I+123,IGAS)**

**LE=I=GPRP(I+124,IGAS)**

**GO TO 520**

**IF (INT.EQ.0) GO TO 530**

**NT=2**

**GO TO 540**

**NT=1**

**CONTINUE**

**LOOK UP ELEMENTS**

**DO 550 I=1,ISC**

**ELEMENT(I)=EPRP(I,II)**

**CMW(I)=EPRP(2,II)**

**LOOK UP SPECIES**

**IAMBIP=1**

**NFIT=0**

**DO 560 I=1,ISS**

**II=II(I)**

**DO 570 K=1,NCS**

**IF (II.EQ.JCS(K)) GO TO 570**

**CONTINUE**

**GO TO 590**

**IJCS(K)=I**

**CONTINUE**

**IF (II.EQ.IATOM) JATOM=I**

**IF (II.EQ.IATOM) JMOL=I**

**HP(I)=SPRP(1,II)**
NELS=SPRP(2,II)+0.1
DO 630 K=1,1,ICS
DO 590 J=1,NELS
IES=SPRP(J+2,II)+0.1
IF (IE(K),EQ,IES) GO TO 600
590 CONTINUE
LPI(J,K)=0
GO TO 630
630 AL=SPRP(J+5,II)
LP=AL
DAL=AL-LP
IF (ABS(DAL).LE.0.5) GO TO 620
IF (DAL.GT.0.) GO TO 610
LP=LP-1
GO TO 620
610 LP=LP+1
LPI(J,K)=LP
IF (I1,EQ,ISATOM,AND,IES,EQ,1) IAMBI=2
620 CONTINUE
TFA(I)=SPRP(9,II)
TFF(I)=SPRP(10,II)
TFNC(I)=SPRP(11,II)
TFC(I)=SPRP(12,II)
TFE(I)=SPRP(13,II)
TFK(I)=SPRP(14,II)
SHJP(I)=SPRP(15,II)
ETAJ(I)=SPRP(16,II)
SUBJ(I)=SPRP(17,II)
THEV(I)=SPRP(18,II)
IF (ETAJ(I),LT,2.9) GO TO 650
DO 640 KE=1,3
640 THEVE(K+1,I)=SPRP(K+40,II)
650 IGM(I)=SPRP(19,II)+0.1
IGJ(I)=SPRP(20,II)+0.1
IF (IGJ(I),EQ,0) NFIT=1
IF (IGJ(I),GE,0) IGJ(II)=IGJ(I)
WRITE (6,1190) I,II
WRITE (6,1190) KE,II
STOP
660 IGMV=IGM(I)
DO 670 KE=1,IGMV
670 GELJ(L,I)=SPRP(L+20,II)
680 CONTINUE
DO 700 I=1,1,ISC
CAPQ(I)=0.
DO 700 J=1,NCS
JJ=JCS(J)
NELS=SPRP(2,JJ)+0.1
DO 590 KE=1,NELS
KK=SPRP(K+2,JJ)+0.1
IF (KK,NE,IE(J)) GO TO 690
CAPQ(I)=CAPQ(I)+OPJ(J)*SPRP(K+5,JJ)
690 CONTINUE
700 CONTINUE
C L3OK UP REACTIONS

REAL 495
REAL 496
REAL 497
REAL 498
REAL 499
REAL 500
REAL 501
REAL 502
REAL 503
REAL 504
REAL 505
REAL 506
REAL 507
REAL 508
REAL 509
REAL 510
REAL 511
REAL 512
REAL 513
REAL 514
REAL 515
REAL 516
REAL 517
REAL 518
REAL 519
REAL 520
REAL 521
REAL 522
REAL 523
REAL 524
REAL 525
REAL 526
REAL 527
REAL 528
REAL 529
REAL 530
REAL 531
REAL 532
REAL 533
REAL 534
REAL 535
REAL 536
REAL 537
REAL 538
REAL 539
REAL 540
REAL 541
REAL 542
REAL 543
REAL 544
REAL 545
REAL 546
REAL 547
REAL 548
REAL 549
DO 780 I=1,ISR
   I=IR(I)
   CAI(I)=RPPR(1,II)
   ETA(I)=RPPF(2,II)
   CEACT(I)=RPPR(3,II)
   GO(I)=RPPR(4,II)
   NRSS=RPPR(5,II)+0,1
   NPSS=RPPR(6,II)+0,1
   DO 770 J=1,ISS
   NRSS=NPSS+0,1
   JRS=RPPR(K+6,II)+0,1
   IF (JRS*NE.*IS(J)) GO TO 710
   XNUIJK(I,J)=RPPR(K+12,II)
   GO TO 720
710 CONTINUE
   XNUIJK(I,J)=0
   CONTINUE

720 CONTINUE
   DO 730 K=1,NPSS
   JPS=RPPR(K+9,II)+0,1
   IF (JPS*NE.*IS(J)) GO TO 730
   XNUIJP(I,J)=RPPR(K+15,II)
   GO TO 740
730 CONTINUE
   XNUIJP(I,J)=0
   CONTINUE
   NK=RPPF(19,II)+0,1
   DO 750 K=1,NK
   ISKUR=RPPR(K+19,II)+0,1
   IF (ISKUR*EQ.*IS(J)) GO TO 760
750 CONTINUE
   KUR(I,J)=0
   GO TO 770
760 KUR(I,J)=1
770 CONTINUE
   IF (IGAS.EQ.0 .OR. NT.EQ.1) GO TO 780
   KTF(I)=TNF(1,INT)
   KTHR(I)=TNF(1,25,INT)
   ITR(I)=TNF(1,50,INT)
   CPAR(1,I)=TNF(1,75,INT)
   CPAR(2,I)=TNF(1,100,INT)
780 CONTINUE
   IF (NT.EQ.1) GO TO 860
   IF (IGAS.EQ.0) GO TO 800
   DO 790 I=1,30
   TLST(I)=TNF(I+125,INT)
790 CONTINUE
   PDM(I)=TNF(I+155,INT)
   BPAR=TNF(186,INT)
   DO 820 I=2,ISS
   IF (LP(I).GE.1) GO TO 810
   ICH(I)=2
   GO TO 820
810 ICH(I)=1
820 CONTINUE
   DO 850 I=1,ISR
   IF (KTF(I).NE.4) GO TO 840
   CONTINUE
DO 830 J=2,ISS
IF (XNU1JP(I+J) .EQ. 0) GO TO 830
IPA(I)=J
GO TO 850
830 CONTINUE
840 IPA(I)=0
850 CONTINUE
860 IZERO=0
SUPGO=IGAS .NE. 0 .AND. IGAS .EQ. IGAS
IF (SUPGO) GO TO 870
WRITE (6,1140)
WRITE (6,1150)
CGMW=C
DO 890 I=1,NCS
J=JCS(I)
NELS=SPRP(2,JJ)
CMW=0
DO 880 K=1,NELS
KK=SPRP(K+2,JJ)
ELNAME(K)=EPRP(1,KK)
NATOM(K)=SPRP(K+5,JJ)
CMW=CMW*NATOM(K)*EPRP(2,KK)
IF (KK .NE. 1) GO TO 880
ELNAME(K)=PLUS
NATOM(K)=NATOM(K)
880 CONTINUE
CMW=CMW+CMW*OPJ(I)
IF (SUPGO) GO TO 890
WRITE (6,1160) I,SPRP(1,JJ),JJ,OPJ(I),CMW,(SY1(K),ELNAME(K),SY2(K))
,1,NATOM(K),K=1,NELS
890 CONTINUE
IF (*NOT.*SUPGO) WRITE (6,1170) CMW
IGASO=IGAS
NOZZO=NOZZLE
ICHANO=ICHAN
IGAS=IGS
IF (XMODP1) 900,900,910
900 XMODP1=1*E20
WRITE (6,1180)
910 XMP=XMODP1*2.54
IF (XMAXI .GT. XMODP1 .AND. NMODPT .GT. 1) GO TO 920
FACMP=1.*E10
GO TO 930
920 FNMP=NMODPT-1
EXPM=1./FNMP
FACMP=(CMAXI/XMODP1)**EXPM
930 CONTINUE
IF (NRECO .LE. 0 .OR. ICASE.GT.1) GO TO 950
DO 940 I=1,NRECO
940 READ (1,TPOUT)
950 CONTINUE
RETURN
C
C
FORMAT (20A4) \( \text{REA} \ 660 \)
FORMAT (1H11) \( \text{REA} \ 661 \)
FORMAT (1H1,55X,20HNATA III CODE OUTPUT/SHORUN NO.,I7,7X,4HCASE,I4\( \text{REA} \ 662 \)
1.12H IN THIS JOB,9X,20A///\( \text{REA} \ 663 \)
FORMAT (3OHINVALID INPUT DATA...NRFLS=110) \( \text{REA} \ 664 \)
1.2H STAGNATION ENTHALPY=F9.0,25H BTU/LB, TOTAL MASS FLOW=F9.0\( \text{REA} \ 665 \)
1.0H RESERVOIR PRESSURE=F9.4,23H ATM, TOTAL MASS FLOW=F9.0\( \text{REA} \ 666 \)
1.5,7H LB/SEC\( \text{REA} \ 667 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 668 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 669 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 670 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 671 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 672 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 673 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 674 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 675 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 676 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 677 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 678 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 679 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 680 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 681 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 682 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 683 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 684 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 685 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 686 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 687 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 688 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 689 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 690 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 691 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 692 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 693 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 694 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 695 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 696 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 697 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 698 \)
1.0H RESERVOIR PRESSURE=F9.4,28H ATM, RESERVOIR TEMPERATURE\( \text{REA} \ 699 \)
SUBROUTINE RESET
SIMULATION OF UNIVAC 1108 TIMING ROUTINES RESET AND TIME

IN=1
GO TO 10
ENTRY TIME(I)
IN=2
10 CALL ACUCPU (ICPU)
GO TO (20..30) IN
20 IZEH0=ICPU
GO TO 40
30 I=IZER0-ICPU
40 RETURN
END

RST  1
RST  2
RST  3
RST  4
RST  5
RST  6
RST  7
RST  8
RST  9
RST 10
RST 11
RST 12
RST 13
SUBROUTINE RESTMP
CALCULATES RESERVOIR TEMPERATURE FROM MASS FLOW DATA.

DO DOUBLE PRECISION AM, BM
LOGICAL ERR, SUPG
REAL ACXM(20), ELMNT(10), HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CIUJ
COMMON AA(22, 24), ELMX(20, 10), CAPX(20)
COMMON AFNTS, AFNM, AMACH, AR, ARBA, ARBB, BZERO

1 C, CAPX, CH, CHA, CLNT, CM, CMA, CRA, CRRB
2 CRRB, CRS, CSTA, CT, CTAP, CTB, CTB, CTMAX, CTMX, CTUJ
3 CTP, CTPL, CR, CX, CXX, CXX, CXX, CXX, CXX, CXX, CXX
4 DELT2, DELTAX, DLGAX, DLQAX, DT, ENT, FLUX, HDLX, PCT
5 PCTEST, PRES, PRSS, PRSTH, PTHC, RHA, RHAP, RHOB, RHOB
6 RHOB, RHOC, RHOP, RHOP, RHOP, RHOP, RHOP, RHOP, RHOP
8 TEST, TEST, TBS, TPS, TP, UP, ZP, ZP

COMMON RE(64), IMT(20), BLBK(31), CAI(64), CAPX(20), RES
1 CCPJ(20), CCCT(20), CGU(20), CHI(64), CHI(20), RES
2 CLUN(64), CLNM(64), CMX(20), ETAI(64), ETAJ(20), RES
3 GJ3(20), PENTGJ(20), PGJ(20), PI(64), PICH(64), RES
4 GJ(64), SAU(20), SODCHI(64), SENT(20), SHJ(20), RES
5 SKL(20), SS(20), TB(30), TFA(20), TFB(20), RES
6 TFD(20), TFG(20), TFV(20), XMJA(20), XNU(64), RES
7 XNU(64, 20), XNUU(64, 20), RES

COMMON IC, I*, INFO, INFO, IP, IRUN, ISC, ICPX, RES
1 ISMC, ISMCN, ISP, ISS, ISSN, ISSP1, ISSP2, ISSP3, ISSP4, RES
2 ISPP, ISPP, ISPP, ISPP, ISPP, ISPP, ISPP, ISPP, ISPP, RES
3 ISX3, ISX3, ISX3, ISX3, ISX3, ISX3, ISX3, ISX3, ISX3, RES
4 ISX4, ISX4, ISX4, ISX4, ISX4, ISX4, ISX4, ISX4, ISX4, RES

COMMON IGJ(20), IGX(20), ITB(5), KUR(64, 20), LPI(10, 20), RES
COMMON AC, ELMNT, HP
DIMENSION AA(22, 24), BTAP(64, 20), CAPQ(31), CCI(20), D吉X(20), GJ(20), SB(64)
1 J(20), SDCH(20), SHAP(20), THEV(20), RES
2 EQUIVALENCE (AA(1, 1), AA(1, 1), BB(1, 1), BB(1, 1), RES
3 (GJ3(1), GJ3(1), GJ3(1), GJ3(1), RES
4 (GJ3(1), GJ3(1), GJ3(1), GJ3(1), RES
5 (GJ3(1), GJ3(1), GJ3(1), GJ3(1), RES

COMMON YRMAIN, YK, H, SUPG, WTSTR, NOTAN
COMMON VMSFL, VMASS, CTMX, TSTOP, IS(20)
COMMON VERROR/ ERR
DATA RNAME, HDR/RESMP/
DIMENSION AM(3, 3), BM(3), F(3), CTSAV(3)
NAMELIST / RSTMP / N, CTSAV, FD1, D2, D3, DCT, DMC, BM, CONST, SM, CMA, DMA, DCTAP
NAMELIST/ RDSTMP2 / N, N2, N3, SMAS, HS, PRESA, CTAP, H, SMP, HO, CTAPP

TLAST, PLAST
DATA TEND, PER, *2*000000/
DATA RGS, *9*1434E7/
IF (ISWSA.EQ.0) WRITE (6, 200)
N=0
IF (ISWSA.EQ.0) GO TO 90
PCCP=PRLSA(2) 003256
CONST=RGS*(VMASS/PCCP)**2
C STANDARD GUESS FOR FIRST POINT
CTAP=10000
RES
IF (N2.GT.20) GO TO 160
H0=H
CALL INIT
CALL INTA
IF (E1H) RETURN
H=CMA*CPH/CMA
IF (ISW5A.NE.0) WRITE (6,RSTD2)
IF (ABS(H-HS).LE.HERR) GO TO 130
IF (N2.GT.1) GO TO 120
CTAP=CTAP
CTAP=CTAP/HS/H
GO TO 110
CTSV=CTAP
CTAP=CTAP+(HS-H)*(CTAP-CTAP)/(HO-H)
CTAPP=CTSV
GO TO 110
130 N3=0
140 N3=N3+1
IF (N3.GT.20) GO TO 160
CALL NMAX
IF (ERR) RETURN
SMP=SMP*SORT(RGAS*CTAP/CMA)
IF (ISW5A.NE.0) WRITE (6,RSTD2)
IF (ABS(SMP-SMASS).LE.SMERR) GO TO 150
PRESA=PRESA*SMASS/SMP
CALL INTA
IF (ERA) RETURN
GO TO 140
150 IF (ABS(CTAP-TLAST).GT.TERR*TLAST) GO TO 100
IF (ABS(PRESA-PLAST).GT.PERR*PLAST) GO TO 100
GO TO 190
160 WRITE (6,210)
IF (ISW2B.EQ.0) GO TO 170
WRITE (6,RSTD2)
GO TO 180
170 WRITE (6,RSTDMP)
180 CALL DUMP (RNAME)
190 RETURN
200 FORMAT (26H1TRACE OF RESTMP OPERATION)
210 FORMAT (30H0CONVERGENCE FAILURE IN RESTMP)
END
SUBROUTINE RNKT
LOGICAL ERR, FAILED, TE
REAL S1, S2, S3, T1, T2, T3, Z
REAL ACOM(30), FLMEN(10), HPI(20)
DOUBLE PREC: SION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22, 24), CDIJ(20, 20), CAPX(20)
COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARB6, BZERO,
1, C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP,
2, CPR, CRS, CSTA, CT, CTA, CR, CTC, CTX, CTM, CTH,
3, CTP, CTPL, CTF, CX, CXL, CXN, CXX, CTXN, DATEST, OIDBTEST, DELT1,
4, DELT2, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT,
5, SCSTEST, PRES, PREA, PRES, PRES, PRDO, PRAP, PRB, RHOB,
6, RHOBAR, RMOC, RHOD, RHPL, RTH, ROBAR, ROBAP, SCPG, SGD,
7, SEN, SHEL, SHS, SHU, SU, SUM, SUMG,
8, TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA
COMMON BE(64), BET(20), BLBK(31), CAT(64), CAPXTH(20)
1, CWPJ(20), CEACT(2), CGL(20), CSMU(20), CHI(64), CHII(20),
2, CLN1(23), CLNP(20), CNS(20), CTAJ(20), ETA(16), ETAJ(20),
3, GJB(20), GERTJ(20), GJ(20), GJ(20), PI(64), PICK(64), QM(20),
4, GQ(64), SAJ(20), SDCH(64), SENT(20), SHJ(20), SHJA(20),
5, SK(20), SSJ(30), TBA(20), TFA(20), TFBJ(20), TF(20),
6, TFD(20), TF(20), TFK(20), THE(20), XMJAT(20), XNUJ(64),
7, COMMON BFA(64, 20), BJ(10, 20), GELJ(10, 20),
8, XNU(64, 20), XNUIJ(64, 20),
COMMON IC, IM, INEO, INEOY, IP, IRUN, ISC, ISCP1,
1, ISPC, ISMCR, ISP, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4,
2, ISW1A, ISW2A, ISWP, ISwp, ISW2B, ISW3, ISW4B, ISW5A,
3, ISW5B, ISW6B, IUPE, IZER, JK, LC, M1, M1T,
4, NIT, NNN, NNS, NOS, NOT, NTST,
COMMON IGJ(20), IGJ(20), ITR(5), KUR(64, 20), LPIJ(20, 10),
COMMON ACOM, ELMENT, HP
DIMENSION AA(22, 24), BTA(64, 20), CAPO(31), CCI(20), DGI(20), GJ(20),
1, SDJG(20), SHAJP(20), SHAP(20), TTHEVP(20),
2, EQUIVALENCE (AA(1, 1), AAA(1, 1)), (BTA(1, 1), BTA(1, 1)),
3, (BLB(11), CAPQ(1)), (CAPXTH(20), CCI(1)), (GJA(1), DGI(1)),
4, (CAPX(6), GJ(1)), (SAJ(1), SBHJ(1)), (SSJ(3), SDGJ(1)),
5, (SHAJ(1), HSHAJP(1)), (TTHE(1), TTHEVP(1)),
COMMON /ERROR/ ERR,
COMMON /TNERK/ SDTE, CTETB, DCHA, CHS, SDCHA, DMAX, IFAIL,
COMMON /ERROR/ FAILED,
COMMON /TNERK/ TNE, CTETB, DCHA, CHS, SDCHA, DMAX, IFAIL,
TE=NT*EQ.1
LIM=ISSP1
IF (.NOT. TE) LIM=LIM+2
QPRE=QDRE
DG*MK=2+QDMAX
IF (ISS*SB*EQ.0) GO TO 10
IST.NT=1
WRITE (6,410) ISTMNT
WRITE (6,RNKDMP)

10 DO 20 J=1,ISS
F(J)=DGJ(J)
GJ1(J)=GJP(J)
GJ(J)=GJB(J)+DGJ(J)*HDELX
20 GJ2(J)=GJ(J)
F1(ISSP1)=DT
GJ1(ISSP1)=CTB
CT=CTB+DT*HDELX
GJ2(ISSP1)=CT
IF (TE) GO TO 30
F1(ISSP2)=DTE
GJ1(ISSP2)=CTB
CTE=CTEB+DTE*HDELX
GJ2(ISSP2)=CT
F1(ISSP3)=CHA
GJ1(ISSP3)=CHB
CHA=CH3+DCHA*HDELX
GJ2(ISSP3)=CHA
30 DO 40 J=1,LIM
IFAIL=J
IF (GJ2(J)) 270,40,40
CCNTINU
IFAIL=J
CALL DERIVS
IF (ERR*OR*FAILED) RETURN
IF (ISS*SB=EQ.0) GO TO 50
IST.NT=2
WRITE (6,410) ISTMNT
WRITE (6,RNKDMP)
50 DO 60 J=1,ISS
F2(J)=CGJ(J)
GJ(J)=GJG(J)+DGJ(J)*HDELX
60 GJ3(J)=GJ(J)
F2(ISSP1)=DT
CT=CTB+DT*HDELX
GJ3(ISSP1)=CT
IF (TE) GO TO 70
F2(ISSP2)=DTE
CTE=CTEB+DTE*HDELX
GJ3(ISSP2)=CTE
F2(ISSP2)=DCHA
CHA=CH3+DCHA*HDELX
GJ3(ISSP3)=CHA
70 DO 80 J=1,LIM
IFAIL=J

SUBROUTINE SHOCK (BMF, DEL, GAM, FL, IERR)

KEEP THIS IMPLICIT STATEMENT IN THE UN. AC VERSION

IMPLICIT REAL*8 (A-H, O-Z)

HARVEY BUSS ROUTINE - OBLIQUE SHOCK IN A PERFECT GAS

**********DEFINITIONS OF ARGUMENTS**********

BMF=FREE STREAM MACH NUMBER

DEL=WEDGE HALF ANGLE (DEGREES)

GAM= RATIO OF SPECIFIC HEATS

FL(1)=SHOCK ANGLE (RADIAN)

FL(2)=MACH NUMBER BEHIND SHOCK

FL(3)=DENSITY RATIO, RHCE(BEHIND)/RHCE(INFINITY)

FL(4)=STATIC PRESSURE RATIO, P(BEHIND)/P(INFINITY)

FL(5)=STATIC TEMPERATURE RATIO, T(BEHIND)/T(INFINITY)

FL(6)=TOTAL PRESSURE RATIO, PT(BEHIND)/PT(INFINITY)

FL(7)= TSTAG/TINF

IEoR=1 NORMAL RETURN, IERR=2 ERROR RETURN, IERR=0 DETACHED SHOCK

DIMENSION FL(7)

KIT=0

DEL=DEL/57.295779

SD=SIN(CDLR)

CD=COS(CDLR)

BM2=0.D*F**2

DM4=0.D*F**2

C1=GAM+1.*0

C2=GAM-1.0

B2=(3*M2+2*0)/8M2=GAM*SD**2

C=(2.*F**2+1.)/8M4+(C1**2/4.0+C2/RM2)*SD**2

C3=2.*B

UG=0.

FU=F+UG*(C+UG*(B+UG))

FP=F+UG*(C+UG*(B+UG))

UN=UG-FU/FPU

IF (DAUS (UG/UN-1.0E-8) 50, 50, 20)

IF (KIT=50) 30, 30, 40

KIT=KIT+1

UG=UN

GO TO 10

WRITE (6,100) IERR=2

RETURN

ALP=E+UN

BETA=UN*ALP+C

DIS=ALP**2-4.0*BETA

IF (DIS*GE-0.0) GO TO 60

WRITE (6,110) IERR=2

RETURN

DIS=SGRT(DIS)

UI=(-ALP+DIS)/2.0

U2=(-ALP-DIS)/2.0

IF (UI-U2) 70, 70, 80

RT=UI
GO TO 90
RT=U2
C
80 ZT= SQRT(RT)
90 CONTINUE
C
FL(1)=ATAN(ZT/SQRT(1-ZT**2))
V1=SM2*RT
C
V2=C1**2*BMA*RT-4*0 *(V1-1*0 )*(GAM*V1+1*0 )
V3=(2*0 *GAM*V1-C2)*(C2*V1+2*0)
C
FL(2)= SQRT(V2/V3)
FL(3)=C1*V1/(C2*V1+2*0)
C
FL(4)=V3/C1**2/V1
C
FL(4)=(2*C*GAM*V1-C2)/C1
C
C4=C2/2*0
C
BME2=FL(2)**2
C5=GAM/C2
C
SMFP=1*0 +C4*BME2
C
SMFP=1*0 +C4*BME2
C
FL(6)=(SMFP/SMFP)**C5*FL(4)
C
FL(7)=SMFP
IERR=1
RETURN
C
100 FORMAT (32H EXCEEDED 50 ITERATIONS IN SHOCK)
110 FORMAT (110H ----SHOCK DETACHED IN CLASSICAL WEDGE CALCULATION. UNSHK 79
1M0DIFIED CHENG-KEMP THEORY IS USED FOR REMAINING ANGLES.)
END
SUBROUTINE SIMO

PURPOSE

OBTAIN SOLUTION OF A SET OF SIMULTANEOUS LINEAR EQUATIONS,

\[ AX = B \]

USAGE

CALL SIMO(A,B,N,KS)

DESCRIPTION OF PARAMETERS

A - MATRIX OF COEFFICIENTS STORED COLUMNWISE. THESE ARE

DESTROYED IN THE COMPUTATION. THE SIZE OF MATRIX A IS

N BY N.

B - VECTOR OF ORIGINAL CONSTANTS (LENGTH N). THESE ARE

REPLACED BY FINAL SOLUTION VALUES, VECTOR X.

N - NUMBER OF EQUATIONS AND VARIABLES.

KS - OUTPUT DIGIT

0 FOR A NORMAL SOLUTION

1 FOR A SINGULAR SET OF EQUATIONS

REMARKS

MATRIX A MUST BE GENERAL.

IF MATRIX IS SINGULAR, SOLUTION VALUES ARE MEANINGLESS.

AN ALTERNATIVE SOLUTION MAY BE OBTAINED BY USING MATRIX

INVERSION (GMPD) AND MATRIX PRODUCT (GMPD).

SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED

NONE

METHOD

METHOD OF SOLUTION IS BY ELIMINATION USING LARGEST PIVOTAL

DIVISOR. EACH STAGE OF ELIMINATION CONSISTS OF INTERCHANGING

ROWS WHEN NECESSARY TO AVOID DIVISION BY ZERO OR SMALL

ELEMENTS.

THE FORWARD SOLUTION TO OBTAIN VARIABLE N IS DONE IN

N STAGES. THE BACK SOLUTION FOR THE OTHER VARIABLES IS

CALCULATED BY SUCCESSIVE SUBSTITUTIONS. FINAL SOLUTION

VALUES ARE DEVELOPED IN VECTOR B, WITH VARIABLE 1 IN B(1),

VARIABLE 2 IN B(2), ..., VARIABLE N IN B(N).

IF NO PIVOT CAN BE FOUND EXCEEDING A TOLERANCE OF 0.001,

THE MATRIX IS CONSIDERED SINGULAR AND KS IS SET TO 1.

TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT

KEEP THIS IMPLIED STATEMENT IN THE UNIVAC VERSION

SUBROUTINE SIMO (A,B,N,KS)

C

KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC VERSION

C

C

SUBROUTINE SIMO (A,B,N,KS)

C

KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC VERSION

C

C

SUBROUTINE SIMO (A,B,N,KS)

C

KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC VERSION
SPECIAL CASE N=1

IF (DABS(A(1)) GE TOL) GO TO 40
B(1)=A(1)
RETURN

FORWARD SOLUTION

DO 90 J=1,N
JY=J+1
JJ=J+N+1
BIGA=D
IT=J-J
DO 30 I=J,N

SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN

IJ=IT+l
IF (DABS(BIGA)-DABS(A(IJ))) GT 30.30
BIGA=A(IJ)
IMAX=I
CONTINUE

TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)

IF (DABS(BIGA)-TOL) LT 40.40.50
KS=1
RETURN

INTERCHANGE ROWS IF NECESSARY

II=J+N*(J-2)
IT=IMAX-J
DO 60 K=J,N
II=II+1
I2=II+IT
SAVE=A(I1)
A(I1)=A(I2)
A(I2)=SAVE

DIVIDE EQUATION BY LEADING COEFFICIENT

A(I1)=A(I1)/BIGA
SAVE=B(IMAX)
B(IMAX)=B(IMAX)/BIGA

ELIMINATE NEXT VARIABLE

IF (J-N) GE 70,100,70
IQS=N*(J-1)
DO 90 IX=JY,N
IXJ=IQS+IX
IT=J-IX

90 CONTINUE
DO 80 JX=JY+N
IXJX=N*(JX-1)+IX
JX=IXJX+1
80 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
90 B(IX)=B(IX)-(B(J)*A(IXJ))
C
BACK SOLUTION
C
100 NY=N-1
IT=N*N
DO 110 J=1,NY
IA=IT-J
IB=N-J
IC=J
DO 110 K=1,J
B(IB)=B(IB)-A(IA)*B(IC)
110 IC=IC-1
RETURN
END
LIM1=2
CTMXC=1000.
DO 40 K=1,4
DO 30 I=1,74
30 TPROP(I,K,1)=0.
40 CONTINUE
DO 90 M=LIM1,LIM
'TT=0
II=1
TV=0.
DO 70 I=1,74
IT=IT+1
IZ=1
IF (?T<LE<NDT(II)) GO TO 60
II=II+1
IT=1
60 TV=TV+DELT_TT(II)
IF (M*EQ*1) TT(I)=TV
CT=TV/CTAP
CALL THERM
TPROP(I,1,M)=XMJAT(J)-SHJA(J)/CT
TPROP(I,2,M)=(SHJ(J)-SHJA(J))*RP/1000.
TF(I,J,M)=CCTP(J)*CRA
1.0DPT(I,J,M)=SFNT(J)*CRA
IF (CT-1.1) 70,50,80
70 CONTINUE
80 CTMXC=1000.
90 CONTINUE
DO 100 I=1,IZ
100 WRITE(6,320) TT(I),((TPROP(I,K,M),K=1,4),M=1,LIM)
110 CONTINUE
RETURN
ENTRY STUNT2
WRITE (6,730)
WRITE (6,340)
WRITE (6,350)
DO 150 L=1,NNKQ
I=ISEQ(L)
NNQi=NNQ(I)
DO 140 J=1,NNQi
IIQJ=I(IJ)
JJQJ=J(JI)
IF (J-1) 120,120,130
120 WRITE (6,360) L,IIQJ(J),((Vv(K,1),K=1,5),SPRP(1,IIQJ),SPRP(1,JJQJ))
GO TO 140
130 WRITE (6,370) SFRP(1,IIQJ),SFRP(1,JJQJ)
140 CONTINUE
WRITE (6,330)
WRITE (6,390)
WRITE (6,350)
MV2=1
LO2=0
DO 190 L=1,NKQ
KK=KQ(L)
190 WRITE (56,390) KKSTU(56)
STU 56
STU 57
STU 58
STU 59
STU 60
STU 61
STU 62
STU 63
STU 64
STU 65
STU 66
STU 67
STU 68
STU 69
STU 70
STU 71
STU 72
STU 73
STU 74
STU 75
STU 76
STU 77
STU 78
STU 79
STU 80
STU 81
STU 82
STU 83
STU 84
STU 85
STU 86
STU 87
STU 88
STU 89
STU 90
STU 91
STU 92
STU 93
STU 94
STU 95
STU 96
STU 97
STU 98
STU 99
STU 100
STU 101
STU 102
STU 103
STU 104
STU 105
STU 106
STU 107
STU 108
STU 109
STU 110
MV1=MV2+1
MV2=MV2+NV(KK)
L01=L02+1
L02=NOC(L)
WRITE (6,390) L,KK,(V(I),I=MV1,MV2)
IF (L02.LT.L01) GO TO 190
DO 150 LQ=L01,L02
I=10(LQ)
J=10(LQ)
IF (LQ=L01) 160,160,170
WRITE (6,400) HP(I),HP(J)
GO TO 190
170 WRITE (6,370) HP(I),HP(J)
180 CONTINUE
190 CONTINUE
IF (ISW1B.GT.0) GO TO 240
WRITE (6,250)
ICARD=ICARD+1
DO 200 I=1,ISS
SAVEC(I)=GJA(I)*CMA
DO 230 I=1,ISS
DO 220 J=1,ISS
WRITE (6,260)
DO 210 K=1,30
TV=1000*K
IF (TV.GT.1.0E+06) GO TO 220
ICARD=ICARD+1
CALL TCONS (TV,PRESA)
WRITE (6,270) TV,(G(L,I,J),L=1,3),J,HP(I),HP(J),ICARD
IF (ISW1A.GT.-1) GO TO 210
PUNCH 270, TV,(G(L,I,J),L=1,3),J,HP(I),HP(J),ICARD
210 CONTINUE
220 CONTINUE
230 CONTINUE
WRITE (6,290)
240 RETURN

C
250 FORMAT (1H1,38X,56HAV AVERAGED PAIR CROSS-SECTIONS AS FUNCTIONS OF TEST
1MPEDATURE)
260 FORMAT (///2X,5HTEMP,8X,4HQ(1),8X,4HQ(2),8X,4HQ(3),5X,7HINDICES,5STU
1X,7HSP ECIES,8X,5HCOUNT)
270 FORMAT (F7.0,F3.E12,5X,13H-13.3X,4H,1H-4A,5X,18)
280 FORMAT (///108M NOTE - CHARGED SPECIES CROSS SECTIONS AT ALL TEMPE
1RATURES ARE BASED ON ELECTRON MOLE FRACTIO N IN RESERVOIR)
290 FORMAT (1H1,40X,22HTHERMAL PROPERTIES OF ,A4,6H (H00=,F10,3,11H KCSTU
1AL/MOLE))
300 FORMAT (1H1,10X,3AQ,18H**PHYSICAL MODEL**,3A6/1H0,8X,1HT,2X,12H(MUSTU
10-H00)RT,9X,5H-H00,12X,2HC,12X,2HSO/29X,9HKCAL/MOLE,2X,12HCAL/MSTU
20LE-DEG,2X,12HCAL/MOLE-DEG)
310 FORMAT (1H3,11X,3AQ,18H**PHYSICAL MODEL**,3A6,12X,3AQ,18H***THERMSTU
10 FIT**,3A6/1H0,8X,1HT,2X,12H(MUC-H00)/RT,9X,5H-H00,12X,2HC,12STU
12X,2HSO/29X,9HKCAL/MOLE,2X,12HCAL/MSTU
34MOLE,2X,12HCAL/MOLE-DEG,2X,12HCAL/MOLE-DEG,29X,9HKCAL/MOLE,2X,12HSTU
15
4CAL/MOLE-DEG, 2X, 12H

CAL/MOLE-DEG)
FORMAT (F10.0, 3F14.3, 10X, 3F14.3)
FORMAT (1H14R, 25H TRANSPORT CROSS SECTION DATA)
FORMAT (60X, 7H (INPUT))
12), 10X, 5HVV(3), 10X, 5HVV(4), 10X, 5HVV(5), 9X, 11H INTERACTION)
FORMAT (7110, 1P5E15.3, 9X, 4A4 - A4)
FORMAT (114X, 4A4 - A4)
FORMAT (50X, 8H (EDITED))
FORMAT (110, 120, 1P5E15.3)
FORMAT (1.1+, 113X, 4A4 - A4)
END

STU 166
STU 167
STU 168
STU 169
STU 170
STU 171
STU 172
STU 173
STU 174
STU 175
STU 176
STU 177-
SUBROUTINE THERM

SPECIES THERMAL PROPERTIES

LOGICAL MIX, FI

REAL ACOM(20), ELMNT(10), MP(20)

DOUBLE PRECISION AAA, AA, CAPX, GJ, CDIJ

COMMON AA(22, 24), CDIJ(20, 10), CAPX(20)

COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO

1 C, CARB, CH, CHA, CLNT, CM, CK, CRA, CRP

2 CPT, CTPL, CTT, CX, CXX, CTMX, CTMX, CTMAX

3 DELT2, DELTA, DLOG, DLOGR, DT, ENT, FLUX, HDELX, PCT

4 SPCTEST, PRES, PRESA, PRESB, PRESH, PRH, RAP, RH, RMB

5 RMOC, RMOB, RHPC, ROBAR, ROBAR, SCPG, SOT, SU

6 SEN, SHPG, SC, SL, SM, SM, SU, SUMG

7 TEST, TESTA, TPRT, TSTOP, UP, ZP, ZPA

8 COMMON RA(64), RE(20), BLBK(31), CAI(64), CAPTH(20), THE

9 COMMON IC, ING, INEO, IP, IRUN, Ispir, ISP, ISP1, Isp2, ISP3, ISP4, THE

10 1 SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, THE

11 3 SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, THE

12 1 SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, THE

13 3 SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, SAI, THE

14 COMMON TC(20), TGM(20), ITB(5), KUR(64, 20), LPIJ(20, 10), THE

15 COMMON ACOM, ELMNT, HP

16 DIMENSION AAA(22, 24), BTA(64, 20), CAPO(31), CCI(20), DGJ(20), GJ(20), SB(20)

17 J(20), SGGJ(20), SHJAP(20), THERP(20)

18 EQUIVALENCE (AA(1, 1), AAA(1, 1), (BTA(1, 1), BTA(1, 1)),

19 (GJ(1, 1), GJ(1, 1), (SJJ(1, 1), SJJ(1, 1), (SS(1, 1), SDDG(1, 1)),

20 (SH(1, 1), SH(1, 1), (THERV(1), THERV(1)))

21 COMMON /POLYAT/ THEV(4, 20)

22 COMMON /POLYAT/ THEV(4, 20)

23 COMMON /POLYAT/ THEV(4, 20)

24 COMMON /POLYAT/ THEV(4, 20)

25 COMMON /POLYAT/ THEV(4, 20)

26 COMMON /POLYAT/ THEV(4, 20)

27 COMMON /POLYAT/ THEV(4, 20)

28 COMMON /POLYAT/ THEV(4, 20)

29 COMMON /POLYAT/ THEV(4, 20)

30 COMMON /POLYAT/ THEV(4, 20)

31 COMMON /POLYAT/ THEV(4, 20)

32 COMMON /POLYAT/ THEV(4, 20)

33 COMMON /POLYAT/ THEV(4, 20)

34 COMMON /POLYAT/ THEV(4, 20)

35 COMMON /POLYAT/ THEV(4, 20)

36 COMMON /POLYAT/ THEV(4, 20)

37 COMMON /POLYAT/ THEV(4, 20)

38 COMMON /POLYAT/ THEV(4, 20)

39 COMMON /POLYAT/ THEV(4, 20)

40 COMMON /POLYAT/ THEV(4, 20)

41 COMMON /POLYAT/ THEV(4, 20)

42 COMMON /POLYAT/ THEV(4, 20)

43 COMMON /POLYAT/ THEV(4, 20)

44 COMMON /POLYAT/ THEV(4, 20)

45 COMMON /POLYAT/ THEV(4, 20)

46 COMMON /POLYAT/ THEV(4, 20)

47 COMMON /POLYAT/ THEV(4, 20)

48 COMMON /POLYAT/ THEV(4, 20)

49 COMMON /POLYAT/ THEV(4, 20)

50 COMMON /POLYAT/ THEV(4, 20)

51 COMMON /POLYAT/ THEV(4, 20)

52 COMMON /POLYAT/ THEV(4, 20)

53 COMMON /POLYAT/ THEV(4, 20)

54 COMMON /POLYAT/ THEV(4, 20)

55 COMMON /POLYAT/ THEV(4, 20)
MIX=TRUE
WWF=0.001*(ZT+500.)
WTHO=1.-WWF
GO TO 50
30 MIX=FALSE,
  IF (ETA(J)) 50,50,70
40 MIX=FALSE,
  IF (CT-CTMAX) 70,50,50
50 IF (E1) GO TO 60
CC

THE THERMO FIT

C
XMJAT(J)=TFA(J)*(1.-CLNT)-TFJ(J)=CT*(TFB(J)+CT*(TFC(J)))/2.+CT*(TFD)

1.)/3.+CT*TFE(J)/4.))+SHJA(J)/CT

SHJ(J)=SHJ(J)+CT*(TFA(J)+CT*(TFB(J)+CT*(TFC(J))+CT*(TFD(J))+CT*TFE)

1.))/)

60 CCPJ(J)=TFA(J)+CT*(2.0*TFB(J)+CT*(3.0*TFC(J)+CT*(4.0*TFD(J)+CT*(5.0

THE TFJ(J)))))

CC

THE 73

IF (XDTMIX) GC TO 300
XMTH=XMJAT(J)
SHTF=SHJ(J)
CCPTF=CCP(J)
STF=SHTF/CT-XMTF

C

THE 78

C

PHYSICAL MODEL

70 CCPJ(J)=0,

THE 82

IF (E1) GO TO 80

THE 83

SHJ(J)=0,

THE 84

XMJAT(J)=0.

THE 85

IF (ETA(J)<LT.3.) GO TO 150

THE 86

C

TRANSFORMATION. ROTATIONAL. AND VIBRATIONAL CONTRIBUTIONS FOR

THE 87

LINEAR TRIATOMIC SPECIES

THE 88

Z1-3.5*CLNT

THE 89

D2 14C K=1.4

THE 90

IF (INEQV/=90.000) GO TO 100

THE 91

Z3=THEVE(K,J)

THE 92

GO TO 110

THE 93

100 Z3=THEVE(K,J)/CT

THE 94

IF (Z3*GT.88.) GO TO 140

THE 95

110 ZA=EXP(Z3)

THE 96

Z5=Z4-1.

THE 97

Z5=THEVE(K,J)/Z5

THE 98

IF (INEQV/=1.0) GO TO 120

THE 99

Z7=Z5/Z5

THE 100

GO TO 130

THE 101

120 Z7=0.

THE 102

Z2=SZ2+Z6/CT

THE 103

ZB=ALOG(Z2/Z5)

THE 104

Z3=Z7*(Z4+Z7)

THE 105

CCPJ(J)=CCPJ(J)+Z9

THE 106

IF (E1) GO TO 140

THE 107

XMJAT(J)=XMJAT(J)-Z8

THE 108

SHJ(J)=SHJ(J)+Z6

THE 109

CONTINUE

THE 110
CCP(J) = CCP(J) + 3.5
IF (E1) GO TO 240
Z1 = SHJ(J)
XMJAT(J) = SAJ(J) - Z1 + SHJA(J)/CT + (1.5 + ETAJ(J)) * CTPL
SHJ(J) = SHJ(J) + 3.5 * CT + SHJA(J)
GO TO 240

C TRANSLATIONAL, ROTATIONAL, AND VIBRATIONAL CONTRIBUTIONS FOR
C MONOTOMIC AND DIATOMIC SPECIES
150 Z1 = ETAJ(J) - 1.0
Z2 = Z1 + 2.5
IF (E1) GO TO 160
XMJAT(J) = 0.
160 CCP(J) = 0.
IF (INEQV) 170, 180, 170
170 Z3 = THEV(J)
GO TO 190
180 Z3 = THEV(J)/CT
IF (Z3 + GT + 38 + ) GO TO 230
190 Z4 = EXP(Z3)
Z5 = Z4 - 1.0
IF (Z5) 200, 230, 200
Z5 = THEV(J)/Z5
IF (INEQV * NE * ZERO) GO TO 210
Z7 = Z3/Z5
GO TO 220
210 Z7 = 0.0
220 Z3 = ALOG(Z4/Z5)
Z9 = Z7 * (Z4 * 77)
CCP(J) = Z1 * Z9
IF (E1) GO TO 230
XMJAT(J) = -(Z1 * Z8)
230 CCP(J) = CCP(J) + Z2
IF (E1) GO TO 240
SHJ(J) = Z2 * CT + SHJA(J) + Z1 * Z6
XMJAT(J) = XMJAT(J) + SHJA(J)/CT - (SAJ(J) + Z2 + CLGT)
240

C ELECTRONIC CONTRIBUTIONS (ALL SPECIES)
S1 = 3.0
S2 = 3.0
S3 = 3.0
4 = 1.5
50 250 M = 1, N
XX = CELJ(M, J) * EXP(-ELJ(M, J)/CT)
S1 = S1 * XX
X = XX * ELJ(M, J)
S2 = S2 * XY
A = XX * ELJ(M, J)
250 S3 = S3 * XX
IF (E1) GO TO 260
XMJAT(J) = XMJAT(J) - ALOG(S1)
SHJ(J) = SHJ(J) + S2 / S1
260 IF (SCLGT + 10 - 37) GO TO 270
A = 0.0
GO TO 280
270 X = S2 + 5.0
B(I,2)=X(I)*Q(2,I,1)
CALL ELCOND (SIGMA,SSIG)
CALL KINT (ZKINT)
CALL KANDMU (ZK)
ZKINT=ZK(1)+ZKINT
VISC=S*ZK(2)*1.C-3
FLEWIS=FLEWIS/ZKINT*IAMBIP
PRF=3167*CPTOT*ZK(2)/ZKINT/CMSAVE
IF (ISWBB.EQ.0) RETURN
C
PRINT TRANSPORT PROPERTY VALUES ONLY IF ISWBB.EQ.0
WRITE (6,100)
WRITE (6,110) T,P,VISC,PRF,SIGMA,FLEWIS,CPTOT CMSAVE
WRITE (6,110) (X(J),J=1,N)
RETURN
C
100 FORMAT (1H )
110 FORMAT (1P12E11.3)
END
SUBROUTINE WEDGE
DOUBLE PRECISION ZETA,CAPGAM,ZP,ZPP,Z,AMD,AND,GAMD,FL
LOGICAL *EDGEM,AXISYM,FINAL
COMMON /ROV/ EGM, ANGLE(10), RADIUS(5), W1, DX, W1X, TWEDGE, WK,
1 NAX, NANG, NRAD, WEDGE, AM1X, ZSW9B
COMMON /MC/ MD
COMMON /AL/ DEEL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIA(2),
1 SW, C, JDIM, RCPNT
COMMON /APPROP/ VISC, PH, SIGMA, FLEWIS
COMMON /OUTS/ FVOUT(35), GJMP(20)
EQUIVALENCE (GAMMA,FVOUT(10)), (AM,FVOUT(8)), (VMUINF,FVOUT(14)),
1 (TINF,FVOUT(2)), (RHINF,FVOUT(5)), (UINF,FVOUT(7)),
2 (PINF,FVOUT(1)), (REP,FVOUT(12)), (GMW,FVOUT(13))
DIMENSION XW(10), PW(10), CW(10), DELSTW(1), VS(10), ZTA(10), UN(2)
DIMENSION FL(7), APR(10), AST(2), ASTO(10), CH(10)
DIMENSION QM(10)
DIMENSION GFA(2), GFAK(2)
DATA UN Marker RNAME/SHWEDGE/, ZERO /0/, ANMIN /1.0E-4/,
1 DATA P1/3.1415927/, AST Marker*,**, ALMIN /0.01/,
DATA AE /1.0/, RGC /8,314E7/, SQTP1 /1.7724539/,
ITH=1
ICH=100
I3=100(ISW9B)
H5=FVOUT(5)/14,40,5*(FVOUT(7)*30,48)**2/4,186E7
H=CPWALL*TWEDGE
H AND H5 ARE IN CAL/GM
HRATIO=H/H5
TTRATIO=TWEDGE/TINF
S=A*SORT(0.5*GAMMA)
R2=2.4*INF*R2F
TREF=TF#(1.0+1.0*HRATIO)/6.
CALL TRANSFrena TREF
C=VISCF/VMUINF*TINF/TFREF#0,067197
THE NUMERICAL FACTOR CONVERTS VISC FROM GM/CM-SEC TO LBM/FT-SEC
TAU0=0.628+1.73*HRATIO
A=(GAMMA+1.0)/2.
A2=A*A
A4=A2*A2
A5=A4*A
A7=A5*A5
G2=GAMMA*GAMMA
G3=GAMMA*G2
WK2=A2*K*K
WK3=A4*K*K
WK4=A5*K*K
EPS=(GAMMA-1.0)/(GAMMA+1.0)
EPS2=EPS*EPS
OF=-C0.3386*PINF*SORT(RGC*TINF/(2.0*PI*GMW))*AE
OFM2=C2.5**2*(GAMMA-0.5*(GAMMA+1.0)*TTRATIO)/(GAMMA-1.0)
/FAC=15.0*A7/3*EPS2/WK4
GFC=WK/(2.0*A2*EPS)
A12=A2*A2
/FA1=C2*A2*GAMMA4*EPS2*AM2/WK2
/FA1=C2*A5*K3*/EPS
CHOF1=C0.332*A*EPS
YSFAC=0.125*G5*K3/A4/\eps
AXM=533.*C*AM2/REPF
XS=12.; C*(\eps*TAU*AM*AM2)**2/REPF
WRITE (6,260) UNITH
WRITE (6,270) XM
IF (ITH*EO=2) WRITE (6,280) XS
ITH=ITH+1
GFRAC(1)=GFC/AM
GFRAC(2)=GAMMA*GFC
DO 260 J=1,NREADLE
TN=2.*READLE(IT)
IF (TN*FOG*6.) TN=1.E-6
EN=AM*SORT(C/(REPF*TN/12.*))
TAUN=TAUN*EN
TAUN2=TAUN*TAUN
TAUN4=TAUN*TAUN*TAUN
TAUN6=TAUN*TAUN*TAUN*TAUN
CHD=CHDFAC*TAUN*EN*TN
GFAK=GFRAC*TN/TAUN4
GFAK2=GFRAC/TAUN2
YSFAC=YSFAC*TN/TAUN4
GFAK1=GFAK1/TAUN2
260 CONTINUE
OME1=1./PFAK
DO 230 IA=1,NANGLE
IF (IT*EO=2) GO TO 30
AND=AM
AND=AM*MAX1(ANGLE, ANMIN)
GAMMA=AM*GAMMA
CALL SHOCK (AM*AND, GAMMA, FL, IERR)
IF (IERR=4) GO TO 20
IAX=IA
GO TO 30
20 APRI=FL(I)
IF (IERR=EO=1) GO TO 30
CALL DUMP (RNAME)
RETURN
30 IF (IA*G=1) ITH=2
FINAL=FALSE
ALPHA=ANGLE(IA)/57.29:8
IF (F4F6) IF (ITH*EO=1) GO TO 40
CAPGAM=GFAK(1)*ALPHA
GO TO 50
CAPGAM=GFAK*SOR*(GAMMA/AM*MAX1(1-AM-1, ZERO))
IF (IT*G=1) GO TO 50
SAS*SS*MAX1(AM*ALMIN)
SSA=SAS**2
EXPV=EXP(-SSAQ)
ERFV=1.+ERF(SSA)
GFAK1=GFRAC1*(CFMC2*(EXPV+SQRTI*SSA*ERFV)-0.5*EXPV)
ITH=0
IAX=1
WRITE (6,300) RADLE(IH),ANGLE(IA),CPGM,EGA
WRITE (6,310) (XW(J),J=1,IT)
WRITE (6,320) (PW(J),J=1,IT)
WRITE (6,330) (CWW(J),ASTG(J),J=1,IT)
WRITE (6,340) (CH(J),J=1,IT)
WRITE (6,350) (DELSW(J),J=1,IT)
IF (IS=.EQ.0) GO TO 220
IF (IC=.EQ.2) GO TO 210
WRITE (6,360) (YT(J),J=1,IT)
IF (ISG=.EQ.1) GO TO 220
IF (FINAL) WRITE (6,380)
IT=S
ITAB=ITAB+1
GO TO 60
CONTINUE
END

GO TO 10
RETURN

FORMAT (18H,43X,46H*CONDITONS ON WEDGE MODELS*"
152X,42X*MODIFIED CHENG-KEMP THEOR)
FORMAT (45H MERGING EFFECTS SIGNIFICANT FOR XW LESS THAN 1P10.28"
74X,42X*INCHES*)
FORMAT (45H STRONG-INTERACTION APPROXIMATION BREAKS DOWN FOR XW GRE"
74X,42X*ATER THAN 1P10.28 INCHES*)
FORMAT (11H0)
FORMAT (31H***LEADING-EDGE RADIUS =,F6.3,25H INCH, ANGLE 
12X,42X*GR0"
6X,42X*DEGREES, CAPITAL GAMMA =,1P10.29H, OMEGA ="
6X,42X*1P10.2"
FORMAT (12H XW (INCHES),9X,1H=,1P10E11.2)
FORMAT (12H PW (ATM),11X,1H=,1P10E11.2)
FORMAT (22H GWW (INCHES/SD FT-SFC) =,1X,10(1P10*2.A))
FORMAT (15H STANTON NUMBER,6X,1H=,1P10E11.2)
FORMAT (14H DELSW (INCH),7X,1H=,1P10E11.2)
FORMAT (12H YT (INCHES),9X,1H=,1P10E11.2)
FORMAT (5H ZETA,15X,1H=,1P10E11.2)
FORMAT (5AX.22H* FREE-MOLECULE LIMIT)
END
SUBROUTINE TOLN(ZETA,CAPGAM,ZZP,ZZP)
SOLUTION OF THE CHENG EQUATION FOR FLOW OVER A WEDGE

C KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC 1103 VERSION

LOGICAL CALLED
DATA CALLED /FALSE/
DATA (A-H,D-O-Z)
NAMELIST/FDMP/ XL,*SOTL,01,02,03,C,05,ZT,DLDT

IF (CALLED) GO TO 10
CALL=TRUE
C1=1.00
C3=1.00/3.00
C29=2.00/3.00
C9=9.00
C45=45.00/3.00
C10=10.00/3.00
C4=4.00
C32=1.600
C5=2.00
C76=7.000/9.00
P1=ZETA*C3
P2=0.00

IF (ZETA GT 1.0*1.0) GO TO 20
C HIGHER ANALYTICAL SOLUTION FOR SMALL ZETA
C1=1.0*2.
Z=1.00*0.06*P1*1.0*0.5000*P1*P2+0.024*ZETA
XL=1.00*(P1+1.0*25075)*P2+0.014212
DLDT=0.057*P12+0.62988*P2+0.09808*P1
GO TO 50

C ITERATIVE SOLUTION BASED ON CHENG/KEMP EXACT SOLUTION FOR GAMMA=0.
XL=0.0+P1+P2

IF (N GT 26) GO TO 40
SOTL=SGRT(XL)
C1=C1+SOTL
C2=C2+SOTL
C3=C3+SOTL
C4=C4+SOTL
P3=DLG(Q1)
ZT=C8**4+C220*Q2+C9*Q2-C463*Q1+Q5*(C103+C4*SOTL+C2*Q5)+C769
Z=C8**4*(SOTL-1.0/C2+C3*X3**C32-Q5)
DLDT=Q17
IF (Q17 NE (ZT-ZOTA)/ZETA<LE,1.0*1.0) GO TO 50
XL=XL-(ZT-ZETA)*DLDT

+1 TO 20

IITE (5,60) ZETA,CAPGAM
APITE (6,60) CAPGAM+2
Z=XL*C300*ZETA
ZC300=DLDT*CG300
Z=DLGT(Z**2+(CAPGAM*ZETA)**2)
RETURN

FORMAT (3CHC30) MANY ITERATIONS IN WESOLN.10X.SH.ZETA=.1PE15.8.10X.
WES 55
SUBROUTINE XSECT
SUBROUTINE XSECT EDITS INPUT CROSS SECTION DATA FOR USE IN XSECT
TRANSFORM CALCULATIONS.

LOGICAL ERR
REAL ACNM(30), ELMNT(10), HP(20)
DOUBLE PRECISION AA, CAPX, CDI
COMMON AA(22, 24), CDI(20, 10), CAPX(20)
COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO,
C, CARR, CHA, CLNT, CM, CMA, CR, CRP,
2 CRRB, CRS, CST, CT, CTAP, CTB, TCC, CMAX, CTMXX,
3 CTP, CTPL, CT, CX, CB, CMAX, DTEST, DTEST, DTEST, DTEST,
4 DELT2, DELTA2, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT,
5 SPCE, PRES, PRESS, PREP, PRNC, RHAP, RHM, RHOB,
6 RHOBAR, RHOC, RHOP, RMPL, RHTH, RDBARX, RDBARP, SCPG, SDT,
7 SEN, SHPG, SC, SL, SL64, SM, SU, SU2, SUMG,
8 TEST, TEST, TPRINT, TSTOP, UP, ZPA
9 COMMON BE(64), BET(20), BBLK(31), CAI(64), CAPXTM(20), XSE
1 COMMON CLCM(20), CEACT(64), CGW(20), CMU(20), CHI(20), XSE
2 COMMON CLNC(64), CLNPI(64), CMW(2), ETAI(64), ETAJ(20), GJA(20), XSE
3 COMMON GIB(20), PERTGJ(20), PGI(20), ICHI(64), OM(20), XSE
4 COMMON OH(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20), XSE
5 COMMON SKL(20), SS(20), TR(30), TFA(20), TFB(20), TFC(20), XSE
6 COMMON TFD(20), TFX(20), TF(20), TFV(20), XMJAT(20), XNU1(64), XSE
7 COMMON BETA(64, 20), ELJ(10, 20), FELJ(10, 20), XSE
8 COMMON XNU1J(64, 20), XNU1JP(64, 20), XSE
9 COMMON IC, IM, INEQ, INEQV, IP, IRUN, ISC, ISCPI, XSE
1 COMMON ISMC, ISMCN, ISR, ISS, ISSP1, ISSP2, ISSP3, ISSP4, XSE
2 COMMON IS1A, ISA1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A, XSE
3 COMMON ISH, ISW6A, ISW6B, IUPO, IE61, IEPO, LC, MI, NBIT, XSE
4 COMMON NIT, NN, NN, NNS, NOS, NTT, NTEST, XSE
5 COMMON IGJ(20), IGM(20), ITB(5), KUR(64, 20), LPIJ(20, 10), XSE
6 COMMON ACOV, ELMNT, HP
7 DIMENSION VCOUL(2, 2, 2), XSE
8 DIMENSION IZ(20) XSE
9 COMMON /TRANS/ T0(3, 20, 20), ZM2(20) XSE
10 COMMON /TRANS?/ KKO(100), NNO(100), ISEQ(100), NNNQ, I(50) XSE
11 COMMON /TRANS?/ IIQ(15, 10), XSE
12 COMMON /TRANS?/ JJQ(15, 10), XSE
13 COMMON /TRANS/ WW(5, 5), XSE
14 COMMON /TRANS/ LSS, IIEC, ID1, ID2, XSE
15 COMMON /TRANS/ V, K(100), Nu(100), IQ(40G), JQ(400), NQK, XSE
16 COMMON /TRANS/ NV(15), N
17 COMMON /LN/ IATCN, IMPL, JATCN, JML
18 COMMON /MS/ MM, CTMNXX, TSTOP, IS(20)
19 COMMON /ERROR/ ERR
20 NAMELIST /XSMP/ Y0, ZM2, I, IIES, IIEC, ID1, ID2, V, K, NQ, IQ, JQ,
1 NKK, NV, N
2 DATA VCOUL/*594.1, 71.3, 41.2, 201.3, 41.2, 201, 16.1, 71.9, 51.6, 200.4*
3 1.3, 1.0, 0.0
4 DATA IZ /*20A*/
5 N=I5
6 IELEC=I5(1)
7 IF (IELEC NE.1) IELEC=0
8 FIND CORRESPONDENCE BETWEEN INPUT SPECIES AND CROSS SECTION DATA.
9 JJ=50
10
DO 30 K=1.50
J=51-K
DO 10 L=1.1,N
IF (IS(L)-J) 10,20,10
10 CONTINUE
I(J)=JJ
JJ=JJ-1
GO TO 30
I(J)=L
30 CONTINUE
C FIND INDICES OF SPECIES TO BE USED IN LEWIS NUMBER CALCULATION.
ID1=MIND(IATOM+IMOL)
ID2=MAX(IATOM+IMOL)
CALL CXSECT (ID1,ID2)
C RE-INDEX CROSS SECTION DATA TO CORRESPOND TO ORDER OF INPUT
C SPECIES. SET Q(I,L,J)=1.0 FOR ALL PAIRS OF SPECIES FOR WHICH
C CROSS SECTION DATA ARE PROVIDED.
DO 50 L=1,N
DO 50 J=1,N
Q(I,J,L)=Q(I,J,L)=0
NQ=1.0
NQ(N)=Q(I,J,L)
NQ(K)=1.0
L=0
50 CONTINUE
NQ=1.0
Q(I,J,L)=NQ
L=0
IF (NQ.EQ.0) GO TO 70
DO 60 KM=1,N
L=KV+1
60 V(KV)=V(KM,N)
70 CONTINUE
NQ=1.0
Q(I,J,L)=NQ
L=0
IF (NQ.EQ.0) GO TO 90
DO 80 KM=1,N
L=1.0
80 CONTINUE
L=1.0
DO 90 JG(KM,N)
90 CONTINUE
C FOR K=9 OR 12, THE V-ARRAY DEPENDS ON THE INDEXING OF THE
C SPECIES. THESE DATA MUST ALSO BE RE-INDEXED.
L=0
100 CONTINUE
CALL CXSECT (L,J)
IF (L,EQ.0) GO TO 110
CALL CXSECT (L,J)
IF (J.LE.N AND LL.LE.N) Q(1,LL,J)=1.
GO TO 105
END
C CROSS SECTION DATA TO DELETE SPECIES NOT REQUIRED FOR
C TRANSPORT CALCULATIONS.
KQ=1
FOR KQ=10 OR 12, IT MAY BE NECESSARY TO COMPUTE CROSS SECTIONS FOR
SPECIES NOT INCLUDED IN TRANSPORT CALCULATIONS. STATEMENTS 120 TO 127
CHECK FOR THIS POSSIBILITY. IF SUCH CROSS SECTIONS ARE FOUND, STATEMENTS
140 TO 300 REVISE THE DATA TO ELIMINATE THEM FROM THE TRANSPORT
CALCULATIONS. STATEMENTS 300 THROUGH 390 THEN PROCEED TO EDIT THE XSE
CROSS SECTION DATA TO REMOVE SPECIES NOT REQUIRED FOR THE
C TRANSPORT CALCULATIONS.
C CALL XSECT (L,MV,LL,J)
IF (L.EQ.0) GO TO 300
IF (J.LE.N AND LL.LE.N) GO TO 120
IF (L.GT=1) LQ1=NO(L-1)+1
LQ2=NO(L)
IF (LQ1 GT LQ2) GO TO 120
DO 130 LG=LQ1,LQ2
130 CONTINUE
GO TO 120
C CROSS SECTIONS ARE REQUIRED FOR THE PAIR (LL,J). FIND UNUSED
DO 150 II=2,N
150 CONTINUE
IF (Q(1,II,JJ)=EQ.0) GO TO 190
NO STORAGE SPACE AVAILABLE. ADD AN ADDITIONAL SPECIES TO CROSS.
NO CROSS SECTION COMPUTATIONS.
IF (N.EQ.20) GO TO 540
N=N+1
JJ=MAX(JJ,LL)
DO 160 J=1,50
160 CONTINUE
DO 170 K=1,50
170 CONTINUE
DO 180 K=1,50
180 CONTINUE
GO TO 540
STORE CROSS SECTIONS FOR PAIR (LL,J) IN LOCATION (II, JJ).
Q(1,II, JJ)=1.
IT IS FIRST NECESSARY TO CONVERT ANY KQ=10 STEPS FOR THE PAIR TO
KQ=12.
NSV=NXK
L=0
L=141
IF (KQ(L),NE,10) GO TO 270
LQ2=NO(L)
IF (L.GT=1) LQ1=NO(L-1)
IF (L01 .GE. L02) GO TO 270

210  LQ= L01
       LG=LQ+1
       IV=10(LQ)
       JV=JG(LQ)
       IF ((L-L .NE. J .AND. (IV .NE. LL .OR. JV .NE. J)) .OR. (IV .NE. LL .AND. JV .NE. J)) XSE 171
1  GO TO 260

C CONVERSION TO K0=12 MUST BE MADE FOR CURRENT STEP:
C FIND LOCATION MM AT WHICH NEW DATA ARE TO BE INSERTED IN V-ARRAY

C
N0= L
LV= 0

220  CALL BXSECT (LV, MM, K, L)
IF (LV .NE. 0) GO TO 220

C REVISE CROSS SECTION DATA

230  MV= MMV
       DO 230 K= MM+1, MMV
       V(MV+4)= V(MV)
       V(MV+2)= IV
       V(MV+3)= JV
       V(MV+4)= JV
       MMV= MM+4
       LQ1= LQ1+1
       LV= NV
       DO 240 K= N0, NV
       KV(LV+1)= KV(LV)
       NV(LV+1)= N0(LV)
       LV= LV-1

240  NSV= NSV+1
       N0(LV+1)= LQ1
       L0= LV+2
       LV= LQ
       DO 250 K= LQ1, LQ
       LV= LV-1
       IQ(LV+1)= IQ(LV)
       J0(LV+1)= J0(LV)
       IQ(LQ1)= IV
       J0(LQ1)= JV
       KV(N0)=12
       CONTINUE

250 IF (L0 .LT. L02) GO TO 210

260 CONTINUE
IF (L0 .LT. NV) GO TO 220

N0= NSV
C CONVERSION OF K0=10 TO K0=12 COMPLETED. NOW REVISE DATA TO CHANGE
C PAIR (L0+J) TO (L0+J)
C
LQ2= N0(N0)
DO 290 LQ= 1, LQ2
IF (IQ(LQ) .NE. LL .OR. J0(LQ) .NE. J) GO TO 280
IQ(LQ)= 11
J0(LQ)= J0
280 CONTINUE
L= L+1

290  CALL BXSECT (L, MV, IV, JV)
IF (L*EQ.0) GO TO 110
IF (IV*NE*LL*OR*JV*NE*J) GO TO 290
IV=II
JV=JJ
GO TO 200
C EDIT CROSS SECTION DATA TO ELIMINATE SPECIES NOT REQUIRED IN
C COMPUTATIONS.
300 DO 310 LQ=1,400
IF (IQ(LQ)*GT*N*OR*JO(LQ)*GT*N) IQ(LQ)=0
310 CONTINUE
NEWL=0
NEWLQ=0
NEWMV=1
MV1=2
NSV=NKQ
LQ=1
KO(100)=0
DO 370 L=1,NSV
K=KO(L)
LQ2=NO(L)
IF (LQ2*LT*LQ1) GO TO 330
DO 320 LQ=LQ1+LQ2
IF (IQ(LQ)*NE*0) NEWLQ=NEWLQ+1
320 CONTINUE
330 LQ1=LQ2+1
IF (L*EQ.10) GO TO 350
IF (NEWLQ*LE*NO(NEWL)) GO TO 360
MV2=MV1+MV(K)-1
DO 340 MV=MV1,MV2
NEWMV=NEWMV+1
340 V(NEWMV)=V(MV)
350 NEWL=NEWL+1
KO(NEWL)=K
NO(NEWL)=NEWLQ
GO TO 370
360 NKQ=NKQ-1
370 MV1=MV1+MV(K)
LL=0
DO 380 LQ=1,400
IF (IQ(LQ)*EQ.0) GO TO 380
LL=LL+1
IQ(LL)=IQ(LQ)
JO(LL)=JO(LQ)
380 CONTINUE
C COMPUTE UNSPECIFIED NEUTRAL-NEUTRAL CROSS SECTIONS FROM EMPIRICAL
C MIXING LAW. COMPUTATIONS ARE PERFORMED AT LAST SPECIFIED STEP
C WITH KQ=10, OR, IF NO STEPS WITH KQ=10 ARE SPECIFIED, AFTER ALL
C SPECIFIED CROSS SECTIONS HAVE BEEN COMPUTED.
C
LL=NKQ+1
DO 390 L=1,NKQ
390 CONTINUE
IF (LL*EQ.10) LL=L
IF (LL*LE*NKQ) GO TO 400
KO(LL)=10
NO(LL)=NO(LL-1)
400  NKQ=NKQ+1
     NN=0
     DO 420 L=1,ISS
     IF (IFLEC*EQ*0) GO TO 410
     IZ(L)=LPI(L+1)
     IF (IZ(L)*NEQ*0) GO TO 420
 410   NN=NN+1
     I(NN)=L
     IF (G(1+L,L)*EQ*0) GO TO 540
 420 CONTINUE
     CALL AXSECT (LL*,NN)
     FOR IONIZED SPECIES, USE EFFECTIVE COULOMB CROSS SECTIONS CHOSEN
     TO FIT TRANSPORT CALCULATIONS OF SPITZER;
     IF (LL*EQ*NKQ*AND*NG(LL)*EQ*NG(LL-1)) GO TO 430
     NKQ=NKQ+1
     NO(NKQ)=NO(NKQ-1)
 430   DO 480 J=1,2
     IE=IELEC
     KK=J
     DO 470 JJ=1,2
     DOP 470 JJ=1,2
     IF (LL*,IE*)EQ*(II*,NE*)AND*(II*,NE*,KK)) GO TO 450
 440   NN=0
     DO 450 L=1,ISS
     II=ABS(IZ(L))
     IF (L*EQ*IE*OR*(II*,NE*)AND*(II*,NE*,KK)) GO TO 450
     NN=NN+1
     I(NN)=L
     CALL AXSECT (NKQ*,NN)
     IF (NO(NKQ)*EQ*NO(NKQ-1)) GO TO 460
     KQ(NKQ)=2
     NKQ=NKQ+1
     NO(NKQ)=NO(NKQ-1)
     V(NEW+V+1)=VCOLA(1, KK, J, JJ)
     V(NEW+V+2)=VCOLA(2, KK, J, JJ)
     NEW+V=NEW+V+2
     FF (KK*,LE*1) GO TO 470
     KK=KK-1
     GO TO 440
 470   IE=0
     CONTINUE
     NN=0
     DO 490 L=1,ISS
     IF (IZ(L)*EQ*0) GO TO 490
     NN=NN+1
     I(NN)=L
     CONTINUE
     CALL AXSECT (NKQ*,NN)
     IF (NO(NKQ)*NEQ*NO(NKQ-1)) GO TO 540
     SET UNSPECIFIED ION-NEUTRAL CROSS SECTIONS EQUAL TO TYPICAL VALUE
     ESTIMATED FOR N-O+ INTERACTION.
     DO 500 L=1,ISS
 500   I(L)=L
     CALL AXSECT (NKQ*,I*)
     IF (NO(NKQ-1)*NEQ*NO(NKQ)) GO TO 510
     NKQ=NKQ-1
     XSE 276
     XSE 277
     XSE 278
     XSE 279
     XSE 280
     XSE 281
     XSE 282
     XSE 283
     XSE 284
     XSE 285
     XSE 286
     XSE 287
     XSE 288
     XSE 289
     XSE 290
     XSE 291
     XSE 292
     XSE 293
     XSE 294
     XSE 295
     XSE 296
     XSE 297
     XSE 298
     XSE 299
     XSE 300
     XSE 301
     XSE 302
     XSE 303
     XSE 304
     XSE 305
     XSE 306
     XSE 307
     XSE 308
     XSE 309
     XSE 310
     XSE 311
     XSE 312
     XSE 313
     XSE 314
     XSE 315
     XSE 316
     XSE 317
     XSE 318
     XSE 319
     XSE 320
     XSE 321
     XSE 322
     XSE 323
     XSE 324
     XSE 325
     XSE 326
     XSE 327
     XSE 328
     XSE 329
     XSE 330
If CNKO GT 100, GO TO 540.

RETURN.

INSUFFICIENT CROSS SECTION DATA AVAILABLE. TERMINATE CASE.

WRITE (66,550), ERR=TRUE.

WRITE (66,540). 

GO TO 530.

FORMAT (1/160) TRANSPORT PROPERTIES OF DESIRED MIXTURE CANNOT BE CALCULATED FROM AVAILABLE DATA. REVISE CROSS SECTION INPUT DATA.

END
APPENDIX

PROGRAMMER'S MANUAL FOR THE NOZFIT CODE

The present appendix documents the programming of the NOZFIT code, a small computer program for generating NATA-type curvefits to nozzle profiles. A user's manual for NOZFIT is presented in Appendix D of Volume II of this report (ref. 2).

Section A.1 of this appendix analyzes the NOZFIT main program and the subroutines used to produce plots of nozzle profile curvefits. Section A.2 is a glossary of Fortran symbols for NOZFIT. Finally, a source listing of NOZFIT and its associated subroutines is presented in Section A.3.

A.1 Analysis of NOZFIT Routines

This section discusses and explains the main program and each subroutine of the NOZFIT code. The purpose of this exposition is to provide an entry into the coding for programmers who wish to analyze errors, make corrections, or introduce modifications into the code.

A.1.1 NOZFIT Main Program

NOZFIT consists of two main sections. The first section computes the parameters of the profile fit from the input data (see Appendix D of Volume II); the second produces printed, punched, and plotted output.

(1) Computation of profile fit parameters. -- The parameters* of the curvefit to the nozzle or channel profile are calculated in the loop D8 140, L = 1, 2. For L = 1, the parameters of the upstream sections are determined; for L = 2, the downstream sections are treated. For each region, the calculations start at the throat and proceed away from the throat. The index K enumerates the profile sections for

*The profile fit parameters and the analytical forms in which they appear are defined in Section 4.3 of Volume I (ref. 1).
each region in the order in which they are determined. The index J labels the sections for the entire nozzle, both up-
stream and downstream, in sequence starting at the upstream end. Finally, JP indexes the downstream boundaries of the profile sections in sequence starting at the upstream end.

The algorithm used to determine the parameters for each section depends upon the input "condition" index, IC\(\text{COND}(J)\).

a. IC\(\text{COND}(J) = 1\)

In each region (upstream or downstream), the first section treated is the one adjacent to the throat. This section is required to be a circular arc convex toward the axis (ISHAPE\(\text(J) = 2\)), and must be specified as a throat section (IC\(\text{COND}(J) = 1\)). For such a throat section, dy/dx must be zero at x = 0. Thus, equation I(122) shows that the abscissa P2 of the circle center must be zero.

\[
P_2 = 0
\]  

(178)

The circle radius is equal to PAR\( (1,J)\):

\[
P_3 = 2.54 \cdot \text{PAR}(1,J)
\]  

(179)

where the numerical factor converts from the inch unit used in the input to the centimeter unit used in the output.

Finally, the ordinate of the circle center is equal to the throat radius plus the circle radius (see figure 28):

\[
P_1 = P_3 + \frac{1}{2} \cdot 2.54 \cdot \text{DTH}
\]  

(180)

where DTH is the throat diameter in inches.

b. IC\(\text{COND}(J) = 2\)

For the condition IC\(\text{COND}(J) = 2\), the current profile section is assumed to be straight (ISHAPE\(\text(J) = 1\)) and tangent to an adjacent circular section nearer the throat. The index of the section nearer the throat is denoted by INDEX. The profile parameters for this section are already known, because of the order which the sections are treated.
Figure 28. Geometry of the Condition ICØND(J) = 1
Since the sections can be either upstream or downstream of the throat, and since the adjacent section INDEX can be either convex (ISHAPE(INDEX) = 2) or concave (ISHAPE(INDEX) = 3) toward the axis, there are four cases to be considered, as illustrated in figure 29. These cases are all treated in one set of formulas with the aid of the following symbols:

\[ S_1 = \begin{cases} 
-1 & \text{upstream} \\
+1 & \text{downstream} 
\end{cases} \quad (181) \]

\[ S_2 = \begin{cases} 
-1 & \text{for circle bottom, ISHAPE(INDEX) = 2} \\
+1 & \text{for circle top, ISHAPE(INDEX) = 3} 
\end{cases} \quad (182) \]

The symbol \( S_1 \) has the same sign as the slope of the current, straight section; \( S_2 \) is negative if the circle center \( C \) is above the line passing through the straight section, positive if the center is below.

The point of tangency of the straight and circular sections is marked by a "1" in each part of figure 29. The coordinates of this point are

\[ X_1 = X_c - S_1 S_2 R \sin \Theta \quad (183a) \]

\[ Y_1 = Y_c + S_2 R \cos \Theta \quad (183b) \]

Here \( R = \text{PARAM}(3, INDEX) \) is the circle radius, and \( \Theta = \text{PAR}(1, J) \) is the angle of inclination of the straight section to the nozzle axis, taken positive. It is easy to show that the internal angle of the triangle at the vertex \( C \) is equal to \( \Theta \). Also \( X_c \) and \( Y_c \) are the coordinates of the circle center in the adjacent circular section nearer the throat:

\[ X_c = \text{PARAM} (2, INDEX) \quad (183c) \]

\[ Y_c = \text{PARAM} (1, INDEX) \quad (183d) \]
Figure 29. Geometry of the Condition IC\&ND(J) = 2
The equation for the straight section is

\[ Y - Y_1 = s_1 \tan \theta \cdot (X - X_1) \]  \hspace{1cm} (184)

Thus, from equation I(121),

\[ P_1 = Y_1 - s_1 X_1 \tan \theta \]  \hspace{1cm} (185)

\[ P_2 = s_1 \tan \theta \]  \hspace{1cm} (186)

c. IC\text{COND}(J) = 3

For the condition IC\text{COND}(J) = 3, the current profile section is assumed to be circular and tangent to an adjoining straight section which is nearer the throat. The abscissa of the point of tangency is assumed to be PAR(2,J) (inches). As illustrated in figure 30, there are again four cases to be considered. As in the preceding case, the quantities \( s_1 \) and \( s_2 \), equations (181) and (182), are used to treat all four cases in a single set of formulas.

In the present case, the coordinates of the point of tangency ("1" in figure 30) are known, since

\[ X_1 = 2.54 \cdot PAR(2,J) \]  \hspace{1cm} (187)

and \( Y_1 \) can be calculated from equation I(121) using the known parameter values for the adjoining straight section nearer the throat. The basic problem is to determine the coordinates \((X_c, Y_c)\) of the circle center. Equations (183) are again applicable, and give

\[ X_c = X_1 + s_1 s_2 R \sin \theta \]  \hspace{1cm} (188a)

\[ Y_c = Y_1 - s_2 R \cos \theta \]  \hspace{1cm} (188b)

From equations I(122) and I(123), \( P_2 = X_c \) and \( P_1 = Y_c \). The radius \( P_3 \) is
Upstream, $ISHAPE(J) = 2$

Downstream, $ISHAPE(J) = 2$

Downstream, $ISHAPE(J) = 3$

Figure 30. Geometry of the Condition $ICOND(J) = 3$
\[ P_3 = 2.54 \cdot \text{PAR}(1,J) \]  
(189)

d. \text{IC\O ND}(J) = 4

The condition \text{IC\O ND}(J) = 4 allows the determination of a circular section which is tangent to two adjacent straight sections (J+1 and J-1). Figure 31 illustrates the four cases which must be considered. The straight section nearer the throat has the index INDEX; the one farther from the throat has the index INDEX2. The angles \( \Theta_1 = \text{PAR}(1,\text{INDEX}) \) and \( \Theta_2 = \text{PAR}(1,\text{INDEX2}) \) are known and the abscissa \( X_1 = 2.5^A \cdot \text{PAR}(2,J) \) of the intersection of the two lines is given. Since the curvefit parameters for the section INDEX have already been determined, the ordinate \( Y_1 \) of the intersection can be calculated from eq. 1(121).

Since the circular section to be determined is tangent to both lines, its center \( C \) lies on the bisector of the angle between the two lines. The length of this bisector, from the intersection point "1" to the circle center "C", is denoted by \( L \) (or XL in the Fortran). Also, the angle between the bisector and either of the lines is denoted by \( \varphi \). This angle is given by

\[ \varphi = \frac{1}{2} (\pi - |\Theta_2 - \Theta_1|) \]  
(190)

since \( (\pi - |\Theta_2 - \Theta_1|) \) is the included angle between the lines. Then, from either of the right triangles with a vertex at \( C \),

\[ L = R / \sin \varphi \]  
(191)

For ISHAPE(J) = 2 (parts (a) and (c) of figure 31), the angle CIA is equal to \( \varphi - \Theta_1 \). For ISHAPE(J) = 3 (parts (b) and (d) of figure 31), angle CIA is equal to \( \varphi - \Theta_2 \). Hence, for ISHAPE(J) = 2,

\[ X_C = X_1 - S_1 \cdot L \cos (\varphi - \Theta_1) \]  
(192a)

\[ Y_C = Y_1 + L \sin (\varphi - \Theta_1) \]  
(192b)

For ISHAPE(J) = 3,

\[ X_C = X_1 + S_1 \cdot L \cos (\varphi - \Theta_2) \]  
(193a)
Figure 31. Geometry of the Condition: $\text{COND}(J) = 4$
\[ Y_c = Y_1 - L \sin (\theta - \theta_2) \] (193b)

For either case, the distance from point "1" to the junction point "P" between the circular section and the straight section nearer the throat is equal to \( L \cos \theta \). The projection of this distance onto the X-axis is \( L \cos \theta \cdot \cos \theta_1 \). Hence, the abscissa of the junction point is

\[ X_p = X_1 - s_1 L \cos \theta \cos \theta_1 \] (194)

It is clear from figure 31 that ISHAPE(J) can be equal to 2 only if \( \theta_2 > \theta_1 \) and ISHAPE(J) can be equal to 3 only if \( \theta_1 > \theta_2 \). If the user inputs a value for ISHAPE(J) violating these conditions, an error message is written and execution is terminated.

(2) Input and Output. -- NATA and its auxiliary programs are designed for operation on either a UNIVAC 1108 or an IBM System 360. So far as \( \text{NOZFIT} \) is concerned, the principal difference between the two computer systems is the word length. The UNIVAC 1108 contains 36-bit words which can store 6 alphanumeric characters. The IBM 360 contains 32- and 64-bit words which can store 4 or 8 alphanumeric characters, respectively. The plot routines used at NASA/JSC on the UNIVAC 1108 and at Avco Systems Division on the IBM 360/75 assume that arrays of alphanumeric information are close-packed (i.e., with no unused characters). To provide correct handling of the information on the "header" card*, \( \text{NOZFIT} \) contains read and write statements based on both the A6 format (for the UNIVAC) and the A4 format (for the IBM 360). The correct input/output statements for alphanumeric information are selected by reference to an indicator \text{IWRDL}, which is set in subroutine FRAMEB (see below). Two versions of FRAMEB are available, one for use on the UNIVAC 1108 at the Johnson Space Center, the other for use on the IBM 360/75 at Avco. Apart from the differences in this one subroutine, \( \text{NOZFIT} \)

*See discussion of \( \text{NOZFIT} \) inputs in Appendix D of Volume II (ref. 2).
and its plot subroutines are fully compatible with both computer systems.

The coding of the printed and punched output of the profile curvefit parameters is straightforward. In the profile calculation (statements 240 to 310), the abscissa $X$ is varied by increments of 0.03 inch from $X = X_{\text{START}}$ to $X = X_{\text{START}} + 6$ inches. At each $X$, the index $K$ of the profile section in which $X$ lies is determined by finding the first $K$ for which the downstream boundary $ATP(K)$ is greater than $X$. The ordinate $Y$ is then computed from equation 1(121), 1(122), or 1(123) according as $ISHAPE(K)$ is equal to 1, 2, or 3. The result of this calculation is in centimeters, and is converted into inches for output.

Finally, if the indicator $PL\text{OTS}$ is .TRUE., the profile is plotted by a call to subroutine $GRAPH$. After the final case in the job ($ENDJOB = .TRUE.$), the entry FRAMB2 of subroutine $FRAMEB$ is called. If the program is being run on the Avco IBM 360/75, FRAMB2 calls a library routine which closes the plot data set.

A.1.2 Subroutine $FRAMEB$

Subroutine $FRAMEB$ contains all of the irreconcilable differences between the UNIVAC 1108 and IBM 360 versions of $N_\text{ZFIT}$. Two versions of this routine are provided (Section A.3), one for each type of computer system.

The IBM 360 version sets the indicator $IW\text{ORDL}$ to 2, to induce $N_\text{ZFIT}$ to use A4 formats in reading and writing alphanumeric information. It also calls the Avco library routine $IDFRMV$ to produce an identification frame preceding the plots of nozzle profiles, if the control variable $PL\text{OTS}$ is .TRUE.. When called through the entry FRAMB2, it calls the Avco library routine $PLTND$ to empty the plot buffer and close the plotting data set.

The UNIVAC 1108 version of $FRAMEB$ sets $IW\text{ORDL}$ to 1, to induce $N_\text{ZFIT}$ to use A6 formats in reading and writing alphanumeric information. It performs no other function.
A plot identification frame is produced automatically on the 1108 computers at NASA/JSC when plots are requested on the JOB card, and the data set is also closed automatically.

A.1.3 Subroutine GRAPH

GRAPH is a general-purpose plotting routine which was developed for use in the auxiliary programs to the NATA code. It uses several lower level plot routines programmed by North American Aviation (NAA) as software for the S-C 4020 high-speed microfilm recorder. The capabilities of GRAPH may be summarized as follows:

1. It produces Cartesian graphs of y versus x from data stored in arrays.

2. The data points can be displayed by up to 10 different plot symbols, or not displayed.

3. The data points can be connected by lines, or not connected. The lines can be continuous, dashed (with user-controlled dash length), dotted, or traced by a string of Charactron symbols.

4. Different data sets can be plotted on the same frame. The points from the different data sets can be represented by different plot symbols and/or connected by lines of different types.

5. A parameter value can be associated with each data set. The curves for different data sets can be labelled with numerical labels and a table displaying the labels and the associated parameter values can be shown in the margin to the right of the graph. Up to three distinct parameters, each with up to 10 values, can be included in a single plot.
The arguments in the calling sequence of GRAPH are defined in the Glossary of Fortran Symbols (Section A.2).

If a table of parameter values is to be printed in the frame, the first call to GRAPH must be preceded by a call to the NAA routine SETMIV (references 3, 17) to provide the required space in the right margin:

(CALL SETMIV (30, 125, 24, 24))

(1) Grid generation. -- The first part of GRAPH, down to statement 110, is executed only when the routine is called with NF \neq 0 to establish the grid for a new frame. In this case, the counter JCURVE for data sets plotted and the counter NPL for parameter tables are initialized to zero, and the NAA routine SMXYV (references 3, 17) is called to establish the logarithmic or linear mode for each axis as indicated by the GRAPH arguments LX, LY. Then the NAA routine DXDYV (references 3, 17) is called for each axis to determine the arguments DX, DY, NNI, MMM, III, JJJ, NXX, NYY of subroutine GRIDIV. If either axis is to be used as a logarithmic scale (LX \neq 0 or LY \neq 0), then the corresponding call to DXDYV is skipped, and DX and DY are set to 1 in accordance with instructions in references 3 and 17.

Beginning at statement 50, the parameters NXY(1) and NXY(2), which determine the numbers of characters to be displayed in the labels of vertical and horizontal grid lines, are established. The corresponding values NXX and NYY provided by DXDYV were found to be unsatisfactory in some applications. For each axis, if \log_{10} of the maximum absolute value to be plotted is greater than 5 or less than -3, the grid-line labels are printed in scientific notation, with three significant digits, by setting NXY(1) = -3. Otherwise, a fixed-point format is used, with a number of characters determined from the magnitude of the maximum value to be plotted.

NXY, the GRAPH arguments III and JJJ are reversed in sign to force placement of the grid line labels outside
of the grid area. Also, if numerical labels for the individual curves are to be displayed, the lower limits \( XL \) and \( YL \) are decreased slightly, and the upper limit \( YU \) is increased slightly, to allow space for the labels within the grid area. Then GRIDIV is called to draw the grid and to establish scaling factors for subsequent use in plotting. Finally, the counter ICURVE for curves in the first family (i.e., for various values of the first parameter, if any) is initialized to zero.

(2) Plotting of a data set. -- The second part of GRAPH plots an individual data set, that is, the data contained in the arrays \( X \) and \( Y \) for a single call to GRAPH. This section of the subroutine extends from statement 110 down to statement 190.

First, the counters ICURVE and JCURVE are each incremented by 1. If ICURVE is equal to 1 after this increase, the counter NPL for parameter tables is also increased by 1. Note that ICURVE is equal to 1 at this point if this is the first call to GRAPH for a new frame, or if ICURVE was set to zero in the calling routine before the current call to GRAPH. Setting ICURVE to 0 in the calling routine is a signal to GRAPH that a new family of curves with a new parameter is being started.

Next, the NAA routines \( \text{NXV} \) and \( \text{NYV} \) are used to initialize \( \text{IX1} \) and \( \text{IY1} \) to the raster coordinates corresponding to \( X(1) \) and \( Y(1) \), respectively. Then the plot symbol to be used in displaying the data points is selected on the basis of the arguments \( NC(1) \) and \( NC(2) \). If \( NC(1) \) is negative, the symbol index NSYM is set equal to ICURVE, so that a new symbol from the standard table in \( \text{POINTB} \) is used for the points on each successive curve in a family of curves. If \( NC(1) \) is positive, NSYM is set equal to \( NC(1) \); this option allows direct user control of the plot symbol. For \( NC(1) = 0 \), the individual points are not plotted.
The DØ loop from statement 140 to statement 180 plots
the data points and draws the connecting lines, if any.
First, subroutine PØINTB (Section A.1.4 of this appendix)
is called to plot the current (ith) data point. Then I2
is set equal to I+1. If LINE(I) is nonzero, subroutine
LINEB (Section A.1.5 of this appendix) is called to draw
the line segment from the ith to the I2th point. The
type of line to be drawn (continuous, dashed, etc.) is
determined by the argument LINE in the call to LINEB.
Then IX1 and IX2 are set equal to the raster coordinates
corresponding to X(I2) and Y(I2), respectively, using the
NAA function routines NXV and NYV, for use in calling
PØINTB the next time around the loop.

When these plotting operations have been completed,
if JCURVE is equal to 1, the NAA routine CHSIZV is called
with arguments (3,3) to restore the normal character
size.* Then subroutine RITE2V (references 3, 17) is called
to write the abscissa and ordinate labels ABSL and ORDL
and the plot title TITL in the bottom, left, and top
margins of the plot, respectively.

(3) Parameter tables and labels. -- If the argument
LL is zero, then parameter tables and curve labelling are
not to be used, and a RETURN is executed at statement 190
in GRAPH. If LL is 1, the parameter tables are produced
but the curve labelling is skipped. If LL is neither 0
nor 1, then the curves are labelled also. To implement
the labelling, the code calls the NAA routines NXV and NYV
to determine the raster coordinates, NXV1 and NYV1, of the
first point on the curve. It then computes the raster
coordinates, IX1 and IY1, of a point which lies 16 raster
units away from the first point of the curve in a direction
opposite to that in which the second point lies. It then
writes the curve label (CURVE = ICURVE) into the plot by
calling the NAA subroutine LABELV (references 3, 17).

* A nonstandard character size is set in PØINTB by calling
CHSIZV(3,2), to obtain plot symbols whose height and width
are approximately equal.
The tables of parameter values are produced by the coding from statement 200 to statement 300. For ICURVE = 1, the code first writes the table heading. The heading consists of a sample of the type of line used for the current family of curves, and a two-line printed specification of the nature of the current parameter. The sample line is produced by calling the entry LINEBR of subroutine LINEB (discussed below). It is horizontal and lies at the raster ordinate IY2. The printed heading is produced by calling the NAA routine PRINTV. Its first line lies 15 raster units below the sample line, and its second line is an additional 20 raster units lower. The tables for different parameters (families of curves) have a vertical separation of 300 raster units.

The body of the parameter table consists of rows each containing the numerical label for the curve (equal to CURVE), a sample of the plot symbol used, and the parameter value. The numerical label is omitted for LL = 1. When it is included, it is produced by calling LABLV. The sample plot symbol is produced by calling PRINTB, and the parameter value by another call to LABLV. The arguments NCHAR and NDMAX of LABLV are determined as functions of IPVM, which is $\log_{10}(PVAL)$ rounded down in magnitude to the nearest integer. If IPVM is greater than 4 or less than -2, NCHAR is set to -3 to give scientific notation with three significant figures. Otherwise, fixed-point notation is used, and NCHAR and NDMAX are adjusted to provide the required number of characters and positions to the left of the decimal point.

A.1.4 Subroutine LINEB

Subroutine LINEB draws a straight line between two specified points (X1, Y1), (X2, Y2). When it is called through its entry LINEBR, the two points are specified in raster coordinates (MX1, MY1), (MX2, MY2). The argument LLINE is a three-element array whose values control the type of line produced and its darkness, as defined in the Glossary of Fortran Symbols (Section A.2).
The raster coordinates of the two points to be connected by a line are denoted, within LINEB, by (NX1, NY1), (NX2, NY2). If entry was through LINEBR, these values are set equal to MX1, etc. If entry was through LINEB, the raster coordinates are determined by calling the NAA subroutines NXV and NYV.

For LINE = LLINE(1) = 1, the points are connected with a continuous line by calling NAA subroutine LINEV. This call is executed IDARK times to obtain the desired darkness.

For LINE > 1, beginning in statement 50, the sine (SINA) and cosine (COSA) of the angle between the vector from point 1 to point 2 and the positive X-axis are computed. Then, for LINE = 2, a dashed line is drawn from point 1 to point 2, with the dash length in millimeters being approximately equal to IPAR = LLINE(3). The distance DEL from the beginning of one dash to the beginning of the next is calculated, in raster units, as 10.8 x IPAR. The numerical coefficient in this formula is obtained by noting that the frame width is 1024 raster units and is 190 mm in hard copy. Thus, 1 mm on the hard copy is equivalent to 5.4 raster units, the dash length HDEL is equal to 5.4 x IPAR, and DEL is twice as large, allowing for the gap between two successive dashes. The number of dashes to be used in the line joining the two points is thus NP = DIST/DEL, where DIST is the separation of the points in raster units. If the two points are too close together in comparison with the specified dash length, integer arithmetic gives NP = 0. This possibility should be borne in mind by the user of GRAPH.

LINEB next computes the increments IDX and IDY in raster coordinates between the beginning and end of a single dash, using the previously determined trigonometric

*This description of the dash length is applicable to hard copy from the Calcomp 890 CRT Plotter at Avco, or to enlargements of microfilm plots produced on the SD 4060 at NASA/JSC to a frame size of 7.5 x 7.5 inches.
functions C@A and SINA. It then calls the NAA subroutine LINE2V (references 3, 17) NP times to draw the dashed line joining the end points specified in the arguments. In the DØ loop, DØ 70 I = 1, NP, the quantities VX1 and WY1 represent the desired starting points for each dash in raster coordinates. These quantities are calculated using floating point arithmetic to avoid the loss of accuracy possible in integer arithmetic, and are rounded down to obtain the integer values NX1, NY1 used in the calls to LINE2V.

For LINE = 3, the line joining the specified end points is produced by plotting a string of Charactron symbols. The symbol to be used is specified by IPAR, which is used as the index in the POINTV character table (references 3, 17). The following list is excerpted from references 3 and 17.

<table>
<thead>
<tr>
<th>IPAR</th>
<th>Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>O</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>4</td>
<td>¥</td>
</tr>
<tr>
<td>5</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>*</td>
</tr>
<tr>
<td>10</td>
<td>H</td>
</tr>
<tr>
<td>15</td>
<td>Ø</td>
</tr>
<tr>
<td>22</td>
<td>Ø</td>
</tr>
<tr>
<td>25</td>
<td>I</td>
</tr>
</tbody>
</table>

The separation of successive points, DEL, is set to 5 raster units if IPAR = 0 (i.e., if the plotting character is a dot), and is set to 15 raster units otherwise. The number of characters to be used in tracing the line is then NP = DIST/DEL. The string of characters is then produced by calling the four-argument version of POINTV (references 3, 17) NP times, in the loop DØ120 I = ., NP. The minus sign on the argument IPAR of POINTV suppresses the plotting of a central dot in each character.
A.1.5 Subroutine POINTB

This routine plots a point at the raster coordinates (IX, IY), using a plot symbol selected from a table ICHAR(I) of ten standard symbols. The index NSYMBL in the calling sequence specifies which symbol is to be used. The symbols are as follows:

<table>
<thead>
<tr>
<th>I</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>+</td>
</tr>
<tr>
<td>3</td>
<td>O</td>
</tr>
<tr>
<td>4</td>
<td>*</td>
</tr>
<tr>
<td>5</td>
<td>H</td>
</tr>
<tr>
<td>6</td>
<td>C</td>
</tr>
<tr>
<td>7</td>
<td>¥</td>
</tr>
<tr>
<td>8</td>
<td>ø</td>
</tr>
<tr>
<td>9</td>
<td>λ</td>
</tr>
<tr>
<td>10</td>
<td>I</td>
</tr>
</tbody>
</table>

These symbols are produced by calling the NAA subroutine RITE2V (references 3, 17). The inversions for I = 7 and 9, and the 90-degree rotation for I = 10, are accomplished using the argument K(NSYMBL) of RITE2V. Before RITE2V is called, a call is made to CHSIZV (references 3, 17) to make the height of the symbols nearly equal to their width instead of 50 percent greater, as is normally the case. The argument IDARK of POINTB and RITE2V controls the number of times the symbol is overdrawn and thus its darkness in the plot.

A.2 Glossary of Fortran Symbols for NOZFIT

This section defines the Fortran symbols used in the NOZFIT main program and in its associated subroutines. The variables are listed in alphanumeric order within each routine, and the routines themselves are in alphabetic order. There are no common variables.
A.2.1 Main Program

ABSL(I) Array containing alphanumeric abscissa label for plots

ACØN Conversion factor (180/π) from radians to degrees

ATP(JP) Axial coordinate at the downstream boundary of the JPth profile section (cm)

CARDS Logical control; .FALSE. value suppresses punched card output

COMMENT(I) Hollerith descriptive data

CTHETA Cos (THETA)

DTH Throat diameter (inch)

DX Increment in X for calculation of profile (inch)

ENDJOB Logical control; .TRUE. value terminates the job at the end of the current case

FISHAP(I) Floating point value representing ISHAPE(I) in punched-card output

FNSECT(I) NSECTS(I) + 0.1 for card output

I DØ index

IC ICØND(J) for the Jth profile section

ICØND(J) Index specifying the type of condition to be used in determining the Jth profile section (see input discussion, Appendix D of Volume II, ref. 2 or Section A.1.1 of the present Appendix)

ICURVE Unused argument of subroutine GRAPH

INCR(L) 0 for L = 1; 1 for L = 2
INDEX

Index of the profile section adjacent to the current (Jth) section and nearer the throat.

INDX2

Index of the profile section adjacent to the current (Jth) section and farther from the throat.

ISGN(L)

1 for L = 1; -1 for L = 2.

ISH

ISHAPE(INDEX).

ISHAPE(J)

Indicator for shape of Jth profile section (see input discussion, Appendix D of Volume II).

ISJ

ISHAPE(K).

IWØRD L

Indicator for number of characters per word on the type of computer system being used:

IWØRD L = 1 6-character word (UNIVAC 1108)

IWØRD L = 2 4-character word (IBM 360)

J

Profile section index.

JP

Index of profile section boundaries.

J1

2K - 1; first J value on the Kth card containing PARAM(I,J) data in the punched output.

J2

J1 + 1; second J value on the Kth card containing PARAM(I,J) data in the punched output.

K

DØ index; profile section index in calculation of profile from curvefit.

KP

Continuation card counter punched in column 6 of output cards.
L Indicator for upstream profile sections (L = 1) and downstream sections (L = 2)

LINE(I) Argument of subroutine GRAPH (see below)

LL Argument of subroutine GRAPH (see below)

NC(I) Argument of subroutine GRAPH (see below)

NOZZLE Profile index for use in NATA

NS NSECTS(L) in computation of profile fit parameters; total number of sections in coding of output

NSML NS-1

NSECTS(L) Number of profile sections upstream (L = 1) and downstream (L = 2) of the throat

ORDL(I) Alphanumeric label for ordinate in plot

PAR(I,J) Input parameter values for the Jth profile section; see discussion of inputs, Appendix D of Volume II

PARAM(I,J) Parameters of curvefit for Jth profile section; see discussion of outputs, Appendix D of Volume II

PHI 0.5 (π - |Θ2-Θ1|); see discussion of ICOND = 4 in Section A.1.1.

PI 3.14159265

PLABL(I) Argument of subroutine GRAPH (see below)

PLOTS Logical control, .FALSE. value suppresses plot output

PV AL Argument of subroutine GRAPH (see below)
R Circle radius for a circular-arc section (cm)

RTHCM Throat radius in centimeters

STHETA Sin (THETA)

S1 -1.0 for \( L = 1 \); +1.0 for \( L = 2 \)

S2 0.0 for ISHAPE(J) = 1; -1.0 for ISHAPE(J) = 2; +1.0 for ISHAPE(J) = 3

S2ARR(ISH) Array containing S2 values for the three shapes

THETA Angle of inclination of a straight section to the nozzle axis (radians)

THETA1 For ICOND = 4, inclination angle of the adjacent section closer to the throat

THETA2 For ICOND = 4, inclination angle of the adjacent section farther from the throat

TITLE(I) Title for plots (Hollerith)

TTHETA Tan(THETA)

X Axial coordinate (cm)

XC X-coordinate of the circle center for the adjacent section nearer the throat

XINCH(I) Axial coordinate (inches)

XL \( R / \sin(\Phi) \); see discussion of ICND = 4 in Section A.1.1

XLØ Lower limit of abscissa in plot

XSTART Left limit on X for calculating nozzle profile (inch)

XUP Upper limit of abscissa in plot
XZERØ  Axial coordinate of start of boundary layer, for use in NATA (cm)

XZERØI  XZERØ expressed in inches

XL  Intermediate X values in calculations of parameters for ICØND = 2, 3, and 4

Y  Profile ordinate (cm)

YC  Y-coordinate of the circle center for the adjacent section nearer the throat

YINCH(I)  Profile ordinate (inches)

YLØ  Lower limit on ordinate in plot

YUP  Upper limit on ordinate in plot

YL  Intermediate Y values in calculations of parameters for ICØND = 2, 3, and 4

A.2.2 Subroutine FRAMEB

BIN  Alphanumeric bin number for plot identification frame (Avco)

IWARDL  Defined above (Main program)

PLOTS  Defined above (Main program)

XMEMØ  Alphanumeric memo number for plot identification (Avco)

XNAME  Alphanumeric submitter name for plot identification frame (Avco)
### A.2.3 Subroutine GRAPH

<table>
<thead>
<tr>
<th><strong>Name</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSL</td>
<td>Array containing 48-character label for abscissa scale</td>
</tr>
<tr>
<td>CURVE</td>
<td>ICURVE expressed as a floating-point value</td>
</tr>
<tr>
<td>DC</td>
<td>Parameter controlling the spacing of grid lines on linear scales; DC = 16 is a normal value. The spacing increases or decreases with the value of DC</td>
</tr>
<tr>
<td>DX</td>
<td>Floating point data increment at which vertical grid lines are displayed</td>
</tr>
<tr>
<td>DY</td>
<td>Floating point data increment at which horizontal grid lines are displayed</td>
</tr>
<tr>
<td>DYL</td>
<td>Amount by which YL is decreased and YU increased to allow space for numerical labels of curves</td>
</tr>
<tr>
<td>HYP</td>
<td>Distance (in raster units) between the first two points of a curve</td>
</tr>
<tr>
<td>I</td>
<td>JØ index</td>
</tr>
<tr>
<td>ICURVE</td>
<td>Counter for the number of curves plotted on the same frame. If this argument is set to zero by the calling program, a new parameter label for a new series of curves is printed in the margin; this feature is used only when two or more families of curves are to be plotted in the same frame.</td>
</tr>
<tr>
<td>IDXV</td>
<td>Interval in raster units between the abscissas of the first two points in a curve</td>
</tr>
<tr>
<td>IDYV</td>
<td>Interval in raster units between the ordinates of the first two points of a curve</td>
</tr>
</tbody>
</table>
IERR

Error indicator in subroutine DXY

(references 3, 17); if a nonzero value of IERR is returned, grid generation is impossible with the given upper and lower limits on X and Y.

III

Argument of subroutine GRID1V (references 3, 17). Every IIIth vertical line of the grid is labelled with a numerical value; a negative III forces placing the labels outside the grid.

IMAG(I)

For I = 1 and 2, log₁₀ of the largest numerical value to be plotted on the abscissa and ordinate, respectively.

IPVM

Log₁₀ of the absolute magnitude of a parameter value (PVAL).

IX1

Abscissa of the location where a curve label is to be printed (raster units).

IY

Ordinate of the location where a parameter value is to be printed in the margin (raster units).

IY1

Ordinate of the location where a curve label is to be printed (raster units).

IY2

Ordinate of the location where a sample of the type of line used for a family of curves is drawn in the margin (raster units).

I2

i + 1

J

DP index

JCURVE

Counter for the total number of curves plotted in a given frame.
**JJJ**

Argument of subroutine GRID1V (references 3, 17). Every JJJth horizontal line of the grid is labelled with a numerical value; a negative JJJ forces placing the labels outside the grid.

**LINE(1)**

Control parameters for lines joining data points, defined as follows:

- **LINE(1)**
  - 0: Do not join points with lines
  - 1: Join points with continuous lines
  - 2: Join points with dashed lines
  - 3: Join points with strings of characters

**LINE(2)**

Number of strikes for line; use 2 for dark lines.

**LINE(3)**

Ignored if LINE(1) is 0 or 1
- Length of dashes in mm if LINE(1) = 2
- Index of character in POINTV table if LINE(1) = 3 (see Procedures Manual, Computation and Analysis Division, MSC, p. 5, c. 7.5.1.127)
  - (Use LINE(3) = 0 with LINE(1) = 3 to produce dotted lines)

**LL**

Control parameter for labelling of curves:
- **LL** = 0: Suppress labelling of curves and printing of parameter values in table at right of grid
- **LL** = 1: Print parameter values and plot symbols but do not label curves
- **LL** = 2: Label curves and print parameter values and plot symbols

**LX**

Argument of subroutine SMXYV (references 3, 17); 0 sets linear mode for abscissa, nonzero log mode.
LY

Argument of subroutine SYXYV (references 3, 17); 0 sets linear mode for ordinate, nonzero log mode

MMM

Argument of subroutine GRIDLV (references 3, 17). Every MMMth horizontal grid line is retraced for emphasis

NC(I)

For I = 1 and 2, control parameters for plotting points:

NC(1) < 0 Use standard table of 10 characters for plotting points, a new character for each successive call to GRAPH for the same family of curves in the same frame (see subroutine PRINTB)

NC(1) = 0 Suppress plotting of points

NC(1) > 0 Use the NC(1)th character in the standard table (see PRINTB)

NC(2) Number of strikes for each symbol plotted; use 1 for light characters, 2 for dark

NCHAR

Argument of subroutine LABLV (see references 3, 17); number of characters to be displayed, including leading blanks and the decimal point, if any

NC1 NC(1)

NC2 NC(2)

NDMAX

Argument of subroutine LABLV (see references 3, 17); maximum number of characters to be displayed to the left of the decimal point
Control parameter for frame advance:

NF = 0  Plot data on same frame
NF = 1  Start a new frame, establish a grid, and plot the data on the new frame

Argument of subroutine RITE2V (references 3, 17); not used in GRAPH

NNN  Argument of subroutine GRID1V (references 3, 17). Every NNNth vertical grid line is retraced for emphasis

Number of data points to be plotted

Counter for families of curves plotted in the same frame

Index of point symbol in standard table (see POINTB)

Abscissa of first point in raster coordinates

Argument of subroutine DXDYV (references 3, 17); not used in GRAPH

For I = 1 and 2, the number of characters to be displayed in the labels of vertical and horizontal grid lines; arguments of subroutine GRID1V (references 3, 17)

Ordinate of first point in raster coordinates

Argument of subroutine DXDYV (references 3, 17); not used in GRAPH

Alphanumeric label for ordinate

For J = 1, 2, a two-line alphanumeric label for the parameter
PVAL Parameter value
TITL(I) Alphanumeric title
X(I) Array containing the abscissas of the data points
XL Lower limit on X for all data to be plotted in a given frame
XU Upper limit on X for all data to be plotted in a given frame
XYMAG(I) For I = 1 or 2, the largest absolute magnitude abscissa or ordinate value to be plotted in a given frame
Y(I) Array containing the ordinates of the data points
YL Lower limit on Y for all data to be plotted in a given frame
YU Upper limit on Y for all data to be plotted in a given frame

A.2.4 Subroutine LINEB
CÔSA Cosine of the angle between the vector to be drawn and the positive X-axis
DEL Twice the dash length in raster units
DIST Distance between the beginning and end of the vector in raster units
HDEL Dash length in raster units
I DØ index over the dashed or plot symbols used to draw the vector
IDARK Number of times lines or symbols are drawn
IDX
Increment in the abscissa for a dash (raster units)

IDY
Increment in the ordinate for a dash (raster units)

IPAR
Length of dashes in millimeters for LINE = 2; integer (NS) for selecting the plot symbol from the table in subroutine P0INTV (references 3, 17) for LINE = 3

J
DØ index for double-strike plotting to obtain dark lines or symbols

LINE
Control parameter for type of line:
LINE = 1 Continuous line
LINE = 2 Dashed line
LINE = 3 String of Charactron symbols

LLINE(I)
LLINE(1) = LINE
LLINE(2) = IDARK
LLINE(3) = IPAR

MX1
Abscissa of the start point in raster units

MX2
Abscissa of the end point in raster units

MY1
Ordinate of the start point in raster units

MY2
Ordinate of the end point in raster units

NP
Number of dashes in vector

NX1
Abscissa of the start point in raster units

NX2
Abscissa of the end point in raster units

NY1
Ordinate of the start point in raster units

NY2
Ordinate of the end point in raster units
RASTER
Logical indicator, .TRUE. for entry through LINEBR, .FALSE. for entry through LINEB

SINA
Sine of the angle between the vector to be drawn and the positive X-axis

SQ
Square of the distance between the start and end points of the vector (raster units)

VX1
Floating point value of NX1

VY1
Floating point value of NY1

X1
Abscissa of the start point

X2
Abscissa of the end point

Y1
Ordinate of the start point

Y2
Ordinate of the end point

A.2.5 Subroutine PØINTB

CHAR(I)
Array of characters for plotting points

IDARK
Argument of RITE2V controlling darkness of plotted symbol (see references 3, 17)

IX
Abscissa of location where point is plotted (raster coordinate)

IY
Ordinate of location where point is plotted (raster coordinate)

K(I)
Argument of RITE2V controlling orientation of plotted symbol; used to present each of the symbols "H", "C", and "Y" in two orientations

NLAST
Argument of RITE2V (references 3, 17); not used in PØINTB
NSYMBL

Index used to select plotting symbol from CHAR array

A.3 Listing of NØZFIT

Presented below are source program listings for the main program of NØZFIT, and for subroutines FRAMEB, GRAPH, LINEB, and POINTB. Listings are provided for both the IBM (Avco) and UNIVAC (NASA/JSC) versions of FRAMEB.
4540N PROGRAM FOR DETERMINING AREA R := 0 PARAMETERS FOR NATA

LOGICAL ENDCB, CARDS, PLOTS
DIMENSION ISHAPE(12), NSECTS(2), PAR(12), DTH, ENDJOB, XSTART,
1 XZEROI, NOZZL, CARDS, PLOTS

DEFINITIONS OF INPUTS

NSECTS(1) NUMBER OF UPSTREAM SECTIONS IN NOZZLE PROFILE
NSECTS(2) NUMBER OF DOWNSTREAM SECTIONS IN NOZZLE PROFILE
NOTE = NSECTS(1) + NSECTS(2) MUST BE < 10
ISHAPE(J) INDEX SPECIFYING SHAPE OF J-TH SECTION (COUNTING FROM
UPSTREAM INLET)

ISHAPE(J) = 1 STRAIGHT SECTION
ISHAPE(J) = 2 CIRCLE BOTTOM
ISHAPE(J) = 3 CIRCLE TOP

PAR(I,J) PARAMETER VALUES FOR J-TH SECTION
 FOR ISHAPE(J) = 1, PAR(I,J) = ANGLE OF INCLINATION TO
NOZZLE AXIS IN DEGREES (POSITIVE VALUE)
 FOR ISHAPE(J) = 2 OR 3, PAR(I,J) = CIRCLE RADIUS IN INCHES

DTH THROAT DIAMETER IN INCHES

ACON=150.*PI

ICOND(J) INDEX SPECIFYING CONDITION DEFINING J-TH SECTION
ICOND(J) = 1 THROAT CONDITION (Y=1.27*DTH AND DY/DX=0
AT X=0)
ICOND(J) = 2 STRAIGHT SECTION (J) IS TANGENT TO ADJACENT
CIRCULAR SECTION NEARER THE THROAT
ICOND(J) = 3 CIRCULAR SECTION (J) IS TANGENT TO ADJACENT
STRAIGHT SECTION (NEARER THE THROAT)
AT X=2.54*PAR(2,J)
ICOND(J) = 4 CIRCULAR SECTION (J) IS USED TO BREAK A
SHAPE AT X=ADJOINING STRAIGHT
SECTIONS WHICH INTERSECT AT X=2.54*PAR(2,J)

XSTART LEFT LIMIT ON X FOR CALCULATING NOZZLE PROFILE (INCH)
XZEROI INLET POSITION IN INCHES ABOVE THE THROAT (NEGATIVE)
NOZZL NOZZLE INDEX FOR NATA
CARDS LOGICAL CONTROL - FALSE VALUE SUPPRESSES PUNCH OUTPUT
PLOTS LOGICAL CONTROL - FALSE VALUE SUPPRESSES PLOT
ENDJOB LOGICAL CONTROL FOR JOB TERMINATION

DIMENSION OPDL(12), ABSL(12), NC(2), LINE(3)
DIMENSION COMMNT(18), XINCH(201), YINCH(201), TITLE(12)
DATA NSECTS, ISHAPE, ICOND /26*, 0/, PAR /24*0/
DATA ABSL /4* 1* X* INCH* 6* "/
DATA OPDL /4* 1* X* INCH* 6* "/
DATA COMMNT /1* COMMNT(1) TITLE(/1) /
DATA ENDJOB /0, TRUE/ INCR /0.1/
DATA S34AR /0*-1*1/
CALL FPAGE (ICRDL, PLOTS)
IF (ICRDL<0) GC TO 20
READ (5,330) (COMMNT(I), I=1, 18)
GO TO 30
READ (3, 340) (CMMXT(I), I=1, 12)
C FIRST 4 CHARACTERS ON THIS CARD SHOULD BE NAME OF FACILITY. FOR USE
IDENTIFICATION IN NATA (DCA, EOS, ETC.)
READ (5, INPUT)
*RITE (6, INPUT)
C ZERO ARRAYS
DO 40 I=1, 12
IF (I*NE.12) ATP(I)=0.
DO 40 J=1, 3
PARX(J, I)=0.
C COMPUTE PROFILE FIT PARAMETERS
DO 140 L=1, 2
C L=1 UPSTREAM
L=2 DOWNSTREAM
NS=NSECTS(L)
DO 130 K=1, NS
IF (L(LEQ2) GO TO 50
S1=1.
J=NS-K+1
ISGN=1
GO TO 60
J=NSECTS(1)+K
ISGN=-1
S1=1.
GO TO (70*20*90*100)+IC
70 IF (ISHAPE(J)*NE.2) GO TO 120
C THR*CAT SECTION (ICOND=1)
PARAM(3, J)=S2*PAR(1, J)
PARAM(2, J)=0.
PARAM(1, J)=PARAM(3, J)+1.27*DTH
ATP(JP)=0,
GO TO 130
80 IF (ISHAPE(J)*NE.1) GO TO 120
IF (ISHAPE(INDEX), NE.2 AND, ISHAPE(INDEX), NE.3) GO TO 120
C CONDITION ICOND=2
R=PARAM(3, INDEX)
THETA=PAR(1, J)/ACON
ISH=ISHAPE(INDEX)
S2=S2ARR(ISH)
XC=PARAM(2, INDEX)
YC=PARAM(1, INDEX)
STHETA=SIN(THETA)
CTHETA=COS(THETA)
THETA=STHETA/CTHETA
X1=XC-S1*S2*R*STHETA
Y1=S2*R*CTHETA+YC
PARAM(1, J)=Y1-S1*X1*THETA
PARAM(2, J)=S1*THETA
PARAM(3, J)=0.
ATP(JP)=X1
GO TO 130
90 IF (ISHAPE(J)*NE.2*AND*ISHAPE(J)*NE.3) GO TO 120
 IF (ISHAPE(INDEX)*NE.1) GO TO 120
 C CONDITION [COND=3
 R=2.*S2*PAR(1,J)
 XI=2.*S4*PAR(2,J)
 JG=ISHAPE(J)
 S2=S2PAR(ISHAPE(J))
 THETA=PAR(1,INDEX)/ACON
 Y1=PAR(1,INDEX)*PAR(2,INDEX)*X1
 PARAM(1,J)=Y1-52*R*COS(THETA)
 PARAM(2,J)=X1+S1*S2*R*SIN(THETA)
 PARAM(3,J)=R
 100 IF (ISHAPE(J)*NE.2*AND*ISHAPE(J)*NE.3) GO TO 120
 IF (ISHAPE(INDEX)*NE.1) GO TO 120
 INDEX2=J*ISGN
 IF (ISHAPE(INDEX2)*NE.1) GO TO 120
 C CONDITION [COND=4
 R=2.*S4*PAR(1,J)
 XI=2.*S4*PAR(2,J)
 THETA=PAR(1,INDEX)/ACON
 THETA=PAR(1,INDEX)/ACON
 PHI=3.*S*PI-AUS(THETA2-THETA1)
 IF (THETA2*GT*THETA1 AND ISHAPE(J).EQ.3) GO TO 120
 IF (THETA2.LT.THETA1 AND ISHAPE(J).EQ.2) GO TO 120
 Y1=PAR(1,INDEX)+PAR(2,INDEX)*X1
 XLR=SIN(PHI)
 ATRPHI=X1-S1*X1*COS(PHI)*COS(THETA1)
 PARAM(3,J)=R
 IF (ISHAPE(J).EQ.3) GO TO 110
 PARAM(2,J)=X1-S1*X1*COS(THETA1)
 PARAM(1,J)=Y1-X1*SIN(THETA1)
 GO TO 133
 110 PARAM(2,J)=X1-S1*X1*COS(THETA2)
 PARAM(1,J)=Y1-X1*SIN(THETA2)
 GO TO 133
 120 WRITE (6,350) J
 CALL EXIT
 130 CONTINUE
 140 CONTINUE
 C OUTPUT
 C PRINTED OUTPUT
 IF (ICRDL.EQ.1) GO TO 150
 WRITE (6,360) (COMMA(1),I=1,18)
 GO TO 160
 150 WRITE (6,370) (COMMA(1),I=1,12)
 160 WRITE (6,380)
 WRITE (6,390)
 NSE=NSECTS(1)+NSECTS(2)
 GO TO J+1,NSE

IF (J=EOQS) GO TO 170
WRITE (6,490) J,ATP(J),PARAM(I,J),I=1,3
GO TO 160
170 CONTINUE
IF (*NOT CARD) GO TO 240
C
PUNCH 0: OUTPUT AND PRINTED CARD IMAGES
RTHEC=1,2,27+DTH
XZC0=2*X4+XZCRI
DC 175 I=1,2
190 FNSECT(I)=NSECTS(I)+G+1
DO 200 I=1,12
200 IF (I,GT,N) ISHAPE(I)=0
WRITE (6,420)
IF (I,GT,N) GO TO 210
PUNCH 520, (COMMNT(I),I=116),NOZZLE
WRITE (6,440) (COMMNT(I),I=116),NOZZLE
GO TO 220
210 PUNCH 81C, (COMMNT(I),I=111),NOZZLE
WRITE (6,430) (COMMNT(I),I=111),NOZZLE
PUNCH 830, NOZZLE,T:TCM,XZFC,(FNSECT(I),I=12),NOZZLE
WRITE (6,460) NOZZLE,T:TCM,XZFC,(FNSECT(I),I=12),NOZZLE
PUNCH 540, (FISHAP(I),I=112),NOZZLF
WRITE (6,440) (FISHAP(I),I=112),NOZZLF
PUNCH 550, (ATP(I),I=16),NOZLIE
WRITE (6,470) (ATP(I),I=16),NOZLIE
PUNCH 550, (ATP(I),I=711),NOZZLE
WRITE (6,490) (ATP(I),I=711),NOZZLE
DC 234 K=1
1612 J1=2*K-1
J2=J1+1
KP=K+4
WRITE (6,490) KP,((PARAM(I,J),I=13),J=J1,J2),NOZZLF
230 PUNCH 570, KP,((PARAM(I,J),I=13),J=J1,J2),NOZZLE
PUNCH 560, COMMNT(I),NOZZLE
WRITE (6,500) COMMNT(I),NOZZLE
C
PRINTOUT OF PROFILE
240 WRITE (6,500)
WRITE (6,500)
X=XC+?3
DC 310 I=1,201
XINCH(I)=X:START+(I-1)*DX
X=3,54*XINCH(I)
NS1=NS1+1
DC 260 J=1,NSM1
KP
IF (ATP(J),GT,X) GO TO 260
250 CONTINUE
K=NS
260 ISJ=ISHAPE(K)
GO TO (270,280,290), ISJ
270 Y=PARAM(1,K)+PARAM(2,K)*X
GO TO 200
260 Y=PARAM(1,K)-SORT(PARAM(3,K)**2-(X-PARAM(2,K)**2)
GO TO 300  
Y=PARAM(1,K)+SORT(PAPAM(3,K)**2-(X-PARAM(2,K))**2)
NOZ 221
310  
YINCH(I)=Y/2.54  
WHITE(6,510) XINCH(I),YINCH(I)
NOZ 222
310  
CONTINUE  
NOZ 223
310  
IF (.NOT.PLOTS) GO TO 320  
NOZ 224
310  
C  
NOZ 225
310  
PLOT OF PROFILE  
NOZ 226
310  
NC(I)=0  
NOZ 227
310  
NC(I)=1  
NOZ 228
310  
LINE(I)=1  
NOZ 229
310  
LINE(I)=2  
NOZ 230
310  
LL=9  
NOZ 231
310  
XLO=XINCH(1)  
NOZ 232
310  
XUP=XINCH(201)  
NOZ 233
310  
YL=0  
NOZ 234
310  
YUP=8  
NOZ 235
310  
CALL GRAPH (1,0,0,201,XINCH,YINCH,"LO",XUP,YLO,YUP,16,NC,LINE,ORDL)
NOZ 236
310  
CALL ADJL,TITLE,LL,"VAL,PLABL,DECURVE"  
NOZ 237
310  
IF (.NOT.ENDJOB) GO TO 10  
NOZ 238
310  
C  
NOZ 239
310  
END FRAM2  
NOZ 240
310  
CALL LEXIT  
NOZ 241
310  
C  
NOZ 242
310  
FORMAT (1AA4)  
NOZ 243
310  
FORMAT (12A6)  
NOZ 244
310  
FORMAT (20H) INCOMPATIBLE DATA FOR SECTION I3  
NOZ 245
310  
FORMAT (1M1,12A6//)  
NOZ 246
310  
FORMAT (1M1,12A6//)  
NOZ 247
310  
FORMAT (2M1,12A6//) NOZ 248
310  
FORMAT (1M1,12A6//)  
NOZ 249
310  
FORMAT (1M1,12A6//)  
NOZ 250
310  
FORMAT (1M1,12A6//)  
NOZ 251
310  
FORMAT (1M1,12A6//)  
NOZ 252
310  
FORMAT (1M1,12A6//)  
NOZ 253
310  
CALL FRAM2  
NOZ 254
310  
FORMAT (1M1,12A6//)  
NOZ 255
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 256
1.20X,4HNOZZ,13  
NOZ 257
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 258
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 259
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 260
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 261
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 262
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 263
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 264
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 265
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 266
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 267
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 268
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 269
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 270
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 271
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 272
310  
FORMAT (7X,7MDATA ZP1,12,2H /F10,.51H,+F10,.51H,1+F6,.1H+F6,.1H0,NZ)
NOZ 273
310  
FORMAT (1F2.20A3)
NOZ 274
310  
END
SUBROUTINE FRAMFB (IWORDL, PLOTS)
LOGICAL PLOTS

FRAMEB FOR AVCO IBM 360/75
PRODUCES IDENTIFICATION FRAME FOR PLOTS AND CLOSES PLOT DATA SET

DIMENSION XNAME(3), XMEMO(2)
DATA XNAME/*"NADE'*/**, XMEMO/**NOZF'*/*, BIN/*29*/
ICHOL = 2
IF (PLOTS) CALL IDFRMV (XNAME, BIN, XMEMO)
RETURN
ENTRY FRAMB2
CALL PLTND
RETURN
END
SUBROUTINE GRAPH (NLX,LY,NP,X,Y,XL,+YL,YU,DC,NC,LINF,ORDL,ABS,LRA
1,TITL,LL,PL,VL,PL,PL,ICUR,)
DIMENSION X(120),Y(120),LINF(3),ORDL(12),ABS(L12),TITL(12),
1,PL,LL(3,2),XY(2),IMAG(2),XYMAG(2),NC(2)
C NOTE - 42 CHARACTERS IN ABSCTION, ORNATE, AND TITLE LABELS
C NOTE - CALL SETMLV (30,125,24,24) BEFORE FIRST CALL TO GRAPH,
C IF PARAMETER TABLE IS TO BE PRINTED
IF (NLX,EQ=0) GO TO 110
JCURVE=0
NPL=0
CALL SMXYV (LX,LY)
IF (LX,EQ=0) GO TO 10
DX=1
NN=1
GO TO 20
10 CALL DXDYV (1,XL,XU,DX,NNN,I1,NXX,DC,1ERR)
20 IF (LY,EQ=0) GO TO 30
DY=1
MM=1
GO TO 40
30 CALL DXDYV (2,YL,YU,DD,MM,3,JJY,NNY,DC,1ERR)
40 IF (LY,EQ=0) GO TO 50
IF (IFK*,EQ=0) GO TO 50
RETURN
50 XSMAG(1)=AMAX1(ABS(XL),ABS(XU))
XSMAG(2)=AMAX1(ABS(YL),ABS(YU))
DO 90 J=1,2
IMAG(I)=ALOG10(XSMAG(I))
IF (IMAG(I)),GT,5,UR,IMAG(I),LTE=-3) GO TO 80
IF (IMAG(I)),GT,5,UR,IMAG(I),LTE=-3) GO TO 80
60 NXY(J)=ABS(IMAG(I))/4
GO TO 90
70 NXY(J)=IMAG(I)+1
GO TO 90
80 NXY(J)=1
GO TO 90
90 CONTINUE
II=-III
JJ=JJJ
IF (LL,LTE=-2) GO TO 100
XL=YL-YL/022*(XU-XL)
DYL=YL/022*(YU-YL)
YU=YU+DYL
100 CALL GRIDYIV (1,XL,XU,YL,YU,DX,YY,NNN,MM,III,JJJ,NXY(1),NXY(2))
ICURVE=0
110 ICURVE=ICURVE+1
IF (ICURVE,EQ=0) NPL=NPL+1
JCURVE=JCURVE+1
I1=NXV(X(1))
I1=NYV(Y(1))
NC2=NC(2)
NC1=NC(1)
IF (NC1),120,140,130
NCHAR=3
CALL LABELV (PVAL,940,1Y,NCHAR,1,NDMAX)
RETURN
C
FORMAT (51H0GRID GENERATION IMPOSSIBLE WITH THE PARAMETERS XL=.1PE
END
SUBROUTINE POINTB (IX, IY, NSYMBL, IDARK)
DIMENSION CHAR(10), K(10)
DATA CHAR /'X', 'Y', 'H', 'C', 'Y', 'Y', 'H'/
DATA K /690, 270, 90, 270, 0/
CALL CHSIZV (3, 2)
CALL RITE2V (IX, IY, 1023, K(NSYMBL), IDARK, 1, 1, CHAR(NSYMBL), NLAST)
RETURN
END
SUBROUTINE LINEB (X1,Y1,X2,Y2,LLINE)     LIN  1
LINE=LLINE(1), IDARK=LLINE(2), IPAR=LLINE(3)  LIN  2
X1 = ABSCISSA OF FIRST POINT                LIN  3
Y1 = ORIGIN OF FIRST POINT                  LIN  4
X2 = ABSCISSA OF SECOND POINT                LIN  5
Y2 = ORIGIN OF SECOND POINT                  LIN  6
LINE = 1 CONNECT POINTS WITH A CONTINUOUS LINE LIN  7
LINE = 2 CONNECT POINTS WITH A DASHED LINE     LIN  8
LINE = 3 CONNECT POINTS WITH A STRING OF CHARACTRON SYMBOLS LIN  9
IF LINE=1 IPAR IS IGNORED                     LIN 10
IF LINE=2 IPAR = LENGTH OF DASHES IN MILLIMETERS LIN 11
IF LINE=3 IPAR = INTEGER (NS) SELECTING PLOT SYMBOL FROM LIN 12
POINTV TABLE                                 LIN 13
IDARK = NUMBER OF TIMES LINE OR SYMBOLS ARE DRAWN LIN 14
LOGICAL RASTER                                LIN 15
DIMENSION LLINF(3)                            LIN 16
RASTER=FALSE                                  LIN 17
GO TO 1C                                      LIN 18
ENTRY LINEBR(MX1,MY1,MX2,MY2,LLINE)           LIN 19
RASTER=TRUE                                   LIN 20
LINE=LLINE(1)                                 LIN 21
IF (LINE=LE=0) RETURN                         LIN 22
IDARK=LLINE(2)                                LIN 23
IPAR=LLINE(3)                                 LIN 24
IF (.NOT.RASTER) GO TO 20                      LIN 25
NX1=MX1                                      LIN 26
NY1=MY1                                      LIN 27
NX2=MX2                                      LIN 28
NY2=MY2                                      LIN 29
GO TO 30                                      LIN 30
NX1=NXV(X1)                                  LIN 31
NX2=NXV(X2)                                  LIN 32
NY1=NYV(Y1)                                  LIN 33
NY2=NYV(Y2)                                  LIN 34
IF (LINE*GT*1) GO TO 50                      LIN 35
DO 40 I=1,IDARK                               LIN 36
CALL LINEV (NX1,NY1,NX2,NY2) RETURN           LIN 37
50 SQ=(NX2-NX1)**2+(NY2-NY1)**2                  LIN 38
DIST=SQRT(50)                                LIN 39
VX1=NX1                                      LIN 40
VY1=NY1                                      LIN 41
COSA=(NX2-NX1)/DIST                           LIN 42
SINA=(NY2-NY1)/DIST                           LIN 43
IF (LINE*GT*2) GO TO 80                      LIN 44
DEL=10*IPAR                                  LIN 45
NP=DIST/DEL                                  LIN 46
HOEL=0*DEL                                   LIN 47
IDX=HOEL*COSA                                LIN 48
IDY=HOEL*SINA                                LIN 49
DO 70 I=1,IPAR                               LIN 50
DO 60 J=1,IDARK                               LIN 51
CALL LINE2V (NX1,NY1,IDX,IDY)                LIN 52
IF (I*EQ*NP) GO TO 70                        LIN 53
VX1=VX1+DEL*COSA                             LIN 54
        END
REFERENCES


