### Title and Subtitle

THE NATA CODE - PROGRAMMER'S MANUAL, Volume III

### Author(s)

W. L. Bade and J. M. Yos

### Performing Organization Name and Address

Avco Systems Division
201 Lowell Street
Wilmington, Massachusetts 01887

### Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Lyndon B. Johnson Space Center
Houston, Texas 77058

### Abstract

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 glossary 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 precoded 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

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2.</td>
<td>Structure and Operation</td>
<td>2</td>
</tr>
<tr>
<td>2.1</td>
<td>Brief Description of the NATA Code</td>
<td>2</td>
</tr>
<tr>
<td>2.2</td>
<td>Subroutines and Linkage</td>
<td>2</td>
</tr>
<tr>
<td>2.3</td>
<td>UNIVAC 1108 Version</td>
<td>12</td>
</tr>
<tr>
<td>2.4</td>
<td>IBM 360 Version</td>
<td>16</td>
</tr>
<tr>
<td>3.</td>
<td>Analysis of Routines</td>
<td>20</td>
</tr>
<tr>
<td>3.1</td>
<td>Main Program</td>
<td>20</td>
</tr>
<tr>
<td>3.2</td>
<td>Subroutine AESØLN</td>
<td>28</td>
</tr>
<tr>
<td>3.3</td>
<td>Subroutine AGSØLN</td>
<td>30</td>
</tr>
<tr>
<td>3.4</td>
<td>Subroutine AXFIT</td>
<td>32</td>
</tr>
<tr>
<td>3.5</td>
<td>Subroutine AXSECT</td>
<td>33</td>
</tr>
<tr>
<td>3.6</td>
<td>Subroutine BLAYER</td>
<td>33</td>
</tr>
<tr>
<td>3.7</td>
<td>Subroutine BLCALL</td>
<td>44</td>
</tr>
<tr>
<td>3.8</td>
<td>Subroutine BXSECT</td>
<td>45</td>
</tr>
<tr>
<td>3.9</td>
<td>Subroutine CØMM</td>
<td>46</td>
</tr>
<tr>
<td>3.10</td>
<td>Subroutine CXSECT</td>
<td>61</td>
</tr>
<tr>
<td>3.11</td>
<td>Subroutine DERIVS</td>
<td>61</td>
</tr>
<tr>
<td>3.12</td>
<td>Subroutine DSMSØL</td>
<td>62</td>
</tr>
<tr>
<td>3.13</td>
<td>Subroutine DUMP</td>
<td>64</td>
</tr>
<tr>
<td>3.14</td>
<td>Subroutine DUMPEX</td>
<td>64</td>
</tr>
<tr>
<td>3.15</td>
<td>Subroutine ELGØND</td>
<td>65</td>
</tr>
<tr>
<td>3.16</td>
<td>Subroutine ELTIME</td>
<td>65</td>
</tr>
<tr>
<td>3.17</td>
<td>Subroutine EPART</td>
<td>66</td>
</tr>
<tr>
<td>3.18</td>
<td>Subroutine EQCALC</td>
<td>67</td>
</tr>
<tr>
<td>3.19</td>
<td>Subroutine EXACT</td>
<td>70</td>
</tr>
<tr>
<td>3.20</td>
<td>Function EXP</td>
<td>73</td>
</tr>
<tr>
<td>3.21</td>
<td>Subroutine FINDX</td>
<td>76</td>
</tr>
<tr>
<td>3.22</td>
<td>Subroutine FRØZEQ</td>
<td>77</td>
</tr>
<tr>
<td>3.23</td>
<td>Subroutine GEØM</td>
<td>91</td>
</tr>
<tr>
<td>3.24</td>
<td>Subroutine GEØMAR</td>
<td>101</td>
</tr>
<tr>
<td>3.25</td>
<td>Subroutine INGAS</td>
<td>103</td>
</tr>
<tr>
<td>3.26</td>
<td>Subroutine INIT</td>
<td>105</td>
</tr>
<tr>
<td>3.27</td>
<td>Subroutine KANDMU</td>
<td>105</td>
</tr>
<tr>
<td>Page</td>
<td>Subroutine</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>106</td>
<td>Subroutine KINT</td>
<td></td>
</tr>
<tr>
<td>107</td>
<td>Subroutine LIST</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>Subroutine MATINV</td>
<td></td>
</tr>
<tr>
<td>108</td>
<td>Subroutine MODEL</td>
<td></td>
</tr>
<tr>
<td>115</td>
<td>Subroutine NEWRAP</td>
<td></td>
</tr>
<tr>
<td>120</td>
<td>Subroutine NEXTMP</td>
<td></td>
</tr>
<tr>
<td>121</td>
<td>Subroutine NØNEQ</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>Subroutine NRMAX</td>
<td></td>
</tr>
<tr>
<td>139</td>
<td>Subroutine ØUT</td>
<td></td>
</tr>
<tr>
<td>140</td>
<td>Subroutine ØUT1</td>
<td></td>
</tr>
<tr>
<td>148</td>
<td>Subroutine PERT</td>
<td></td>
</tr>
<tr>
<td>149</td>
<td>Function PI$IIIEG</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>Subroutine PRØP</td>
<td></td>
</tr>
<tr>
<td>150</td>
<td>Subroutine PRTA</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>Subroutine PUTQIN</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>Subroutine QCØUL</td>
<td></td>
</tr>
<tr>
<td>155</td>
<td>Subroutine QEX</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>Subroutine QEXP</td>
<td></td>
</tr>
<tr>
<td>156</td>
<td>Subroutine QINTRP</td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>Subroutine QIJ</td>
<td></td>
</tr>
<tr>
<td>157</td>
<td>Subroutine QMIX</td>
<td></td>
</tr>
<tr>
<td>158</td>
<td>Subroutine QREPP</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Subroutine QSAME</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Subroutine QTAB</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Subroutine Q1L</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Subroutine Q12</td>
<td></td>
</tr>
<tr>
<td>159</td>
<td>Subroutine Q1L2</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>Subroutine Q13</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>Subroutine Q14</td>
<td></td>
</tr>
<tr>
<td>160</td>
<td>Subroutine RADIUS</td>
<td></td>
</tr>
<tr>
<td>161</td>
<td>Subroutine READ</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>Subroutine RESET</td>
<td></td>
</tr>
<tr>
<td>166</td>
<td>Subroutine RESTMP</td>
<td></td>
</tr>
<tr>
<td>174</td>
<td>Subroutine RNKT</td>
<td></td>
</tr>
<tr>
<td>178</td>
<td>Subroutine SHØCK</td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>Subroutine SIMQ</td>
<td></td>
</tr>
<tr>
<td>181</td>
<td>Subroutine STUNLS</td>
<td></td>
</tr>
<tr>
<td>186</td>
<td>Subroutine THERM</td>
<td></td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS
(Continued)

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.65 THR@T</td>
<td>188</td>
</tr>
<tr>
<td>3.66 TRANSPI</td>
<td>188</td>
</tr>
<tr>
<td>3.67 WEDGE</td>
<td>192</td>
</tr>
<tr>
<td>3.68 WESØLN</td>
<td>193</td>
</tr>
<tr>
<td>3.69 XSECT</td>
<td>195</td>
</tr>
</tbody>
</table>

4. Glossary of Fortran Symbols ...................................... 212

#### 4.1 Unlabelled Common ............................................. 212

#### 4.2 Common /AECEØM/ ............................................ 224

#### 4.3 Common /AREA/ .............................................. 224

#### 4.4 Common /AVG/ ............................................... 225

#### 4.5 Common /BL/ ................................................ 225

#### 4.6 Common /BLNE/ .............................................. 226

#### 4.7 Common /BLØUT/ ............................................ 227

#### 4.8 Common /BLRAD/ ............................................ 227

#### 4.9 Common /CERRØR/ ........................................... 227

#### 4.10 Common /CHAN/ ............................................. 227

#### 4.11 Common /ÇØLDSP/ ........................................... 228

#### 4.12 Common /ÇØNVRT/ .......................................... 228

#### 4.13 Common /DRV/ ............................................... 229

#### 4.14 Common /BELEM/ ............................................ 229

#### 4.15 Common /ELEM/ ............................................. 230

#### 4.16 Common /EQC/ ............................................. 230

#### 4.17 Common /EQC2/ ............................................ 230

#### 4.18 Common /ERRØR/ ........................................... 231

#### 4.19 Common /GLIM2/ ............................................ 231

#### 4.20 Common /INGNE/ ............................................ 231

#### 4.21 Common /LN/ ............................................... 231

#### 4.22 Common /MASST2/ .......................................... 231

#### 4.23 Common /MDXT/ ............................................. 232

#### 4.24 Common /MØDPBAR/ ......................................... 233

#### 4.25 Common /MØDPT/ ............................................ 234

#### 4.26 Common /MØDP2/ ............................................ 234

#### 4.27 Common /NEQ/ ............................................... 235

#### 4.28 Common /NEWMP/ ............................................ 235

#### 4.29 Common /NØZ2/ ............................................. 235

#### 4.30 Common /ØUTMØD/ .......................................... 236

---

-vii-
<p>| 4.31  | Common /OUTPUT/ | 237 |
| 4.32  | Common /PLYAT/ | 239 |
| 4.33  | Common /RDLIST/ | 239 |
| 4.34  | Common /RDMAIN/ | 239 |
| 4.35  | Common /RDMD/ | 240 |
| 4.36  | Common /RDUUT/ | 240 |
| 4.37  | Common /RDT/ | 240 |
| 4.38  | Common /RDWEDG/ | 240 |
| 4.39  | Common /REA/ | 241 |
| 4.40  | Common /READAT/ | 242 |
| 4.41  | Common /RENE/ | 245 |
| 4.42  | Common /RESPRP/ | 245 |
| 4.43  | Common /SPEC/ | 246 |
| 4.44  | Common /SS/ | 247 |
| 4.45  | Common /STAG/ | 247 |
| 4.46  | Common /STEPS/ | 247 |
| 4.47  | Common /SWITCH/ | 247 |
| 4.48  | Common /TAPUT/ | 248 |
| 4.49  | Common /TEMPY/ | 248 |
| 4.50  | Common /THRT/ | 248 |
| 4.51  | Common /TNCE/ | 249 |
| 4.52  | Common /TNDC/ | 249 |
| 4.53  | Common /TNE/ | 250 |
| 4.54  | Common /TNEQ/ | 250 |
| 4.55  | Common /TNERK/ | 251 |
| 4.56  | Common /TNEN/ | 253 |
| 4.57  | Common /TRAN/ | 254 |
| 4.58  | Common /TRANJ/ | 254 |
| 4.59  | Common /TRANS/ | 254 |
| 4.60  | Common /TRANS2/ | 255 |
| 4.61  | Common /TRANS3/ | 256 |
| 4.62  | Common /TRANS4/ | 256 |
| 4.63  | Common /TRANS5/ | 256 |
| 4.64  | Common /TRANS7/ | 257 |
| 4.65  | Common /TRANS8/ | 257 |
| 4.66  | Common /TRANV/ | 257 |
| 4.67  | Common /TRPRP/ | 258 |
| 4.68  | Common /WMD/ | 258 |</p>
<table>
<thead>
<tr>
<th>Page</th>
<th>Subroutine/Function</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 AES$LN</td>
<td>260</td>
</tr>
<tr>
<td>4.71</td>
<td>Subroutine AGS$LN</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>27</td>
</tr>
<tr>
<td>4.80</td>
<td>Subroutine DSMSøL</td>
<td>27</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 FR$SEQ</td>
<td>280</td>
</tr>
<tr>
<td>4.91</td>
<td>Subroutine GEØM</td>
<td>284</td>
</tr>
<tr>
<td>4.92</td>
<td>Subroutine GEØMAR</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 ØUT1</td>
<td>301</td>
</tr>
<tr>
<td>Reference</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.105</td>
<td>Subroutine PERT</td>
<td>304</td>
</tr>
<tr>
<td>4.106</td>
<td>Function PIØMEG</td>
<td>305</td>
</tr>
<tr>
<td>4.107</td>
<td>Subroutine PRØP</td>
<td>305</td>
</tr>
<tr>
<td>4.108</td>
<td>Subroutine PERTA</td>
<td>305</td>
</tr>
<tr>
<td>4.109</td>
<td>Subroutine PUTQIN</td>
<td>305</td>
</tr>
<tr>
<td>4.110</td>
<td>Subroutine QCØUL</td>
<td>306</td>
</tr>
<tr>
<td>4.111</td>
<td>Subroutine QEX</td>
<td>306</td>
</tr>
<tr>
<td>4.112</td>
<td>Subroutine QEXP</td>
<td>307</td>
</tr>
<tr>
<td>4.113</td>
<td>Subroutine QINTRP</td>
<td>308</td>
</tr>
<tr>
<td>4.114</td>
<td>Subroutine QLJ</td>
<td>308</td>
</tr>
<tr>
<td>4.115</td>
<td>Subroutine QMTX</td>
<td>309</td>
</tr>
<tr>
<td>4.116</td>
<td>Subroutine QREPP</td>
<td>309</td>
</tr>
<tr>
<td>4.117</td>
<td>Subroutine QSAME</td>
<td>309</td>
</tr>
<tr>
<td>4.118</td>
<td>Subroutine QTAB</td>
<td>310</td>
</tr>
<tr>
<td>4.119</td>
<td>Subroutine Q11</td>
<td>310</td>
</tr>
<tr>
<td>4.120</td>
<td>Subroutine Q12</td>
<td>311</td>
</tr>
<tr>
<td>4.121</td>
<td>Subroutine Q13</td>
<td>311</td>
</tr>
<tr>
<td>4.122</td>
<td>Subroutine Q14</td>
<td>312</td>
</tr>
<tr>
<td>4.123</td>
<td>Subroutine RADIUS</td>
<td>312</td>
</tr>
<tr>
<td>4.124</td>
<td>Subroutine READ</td>
<td>313</td>
</tr>
<tr>
<td>4.125</td>
<td>Subroutine RESET</td>
<td>317</td>
</tr>
<tr>
<td>4.126</td>
<td>Subroutine RESTMP</td>
<td>317</td>
</tr>
<tr>
<td>4.127</td>
<td>Subroutine RNKT</td>
<td>319</td>
</tr>
<tr>
<td>4.128</td>
<td>Subroutine SHØCK</td>
<td>322</td>
</tr>
<tr>
<td>4.129</td>
<td>Subroutine SIMQ</td>
<td>324</td>
</tr>
<tr>
<td>4.130</td>
<td>Subroutine STUNTS</td>
<td>325</td>
</tr>
<tr>
<td>4.131</td>
<td>Subroutine THERM</td>
<td>327</td>
</tr>
<tr>
<td>4.132</td>
<td>Subroutine THRØAT</td>
<td>329</td>
</tr>
<tr>
<td>4.133</td>
<td>Subroutine TRANSP</td>
<td>329</td>
</tr>
<tr>
<td>4.134</td>
<td>Subroutine WEDGE</td>
<td>330</td>
</tr>
<tr>
<td>4.135</td>
<td>Subroutine WESØLN</td>
<td>337</td>
</tr>
<tr>
<td>4.136</td>
<td>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)

Appendix - Programmer's Manual for the NOZFIT Code .... 576

A.1 Analysis of NOZFIT Routines ....................... 576
A.2 Glossary of Fortran Symbols for NOZFIT .......... 594
A.3 Listing of NOZFIT .................................... 608

References ............................................... 624
# LIST OF ILLUSTRATIONS

<table>
<thead>
<tr>
<th>Illustration</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Overlay Structure for UNIVAC 1108 Version of NATA</td>
<td>14</td>
</tr>
<tr>
<td>2.</td>
<td>Control Cards for Generating a NATA PCF Tape on the UNIVAC 1108</td>
<td>15</td>
</tr>
<tr>
<td>3.</td>
<td>Control Cards for Running NATA on the UNIVAC 1108</td>
<td>15</td>
</tr>
<tr>
<td>4.</td>
<td>Control Cards for Running NATA on the IBM 360</td>
<td>19</td>
</tr>
<tr>
<td>5.</td>
<td>Flowchart of NATA Main Program</td>
<td>21</td>
</tr>
<tr>
<td>6.</td>
<td>Flowchart of Subroutine BLAYER</td>
<td>35</td>
</tr>
<tr>
<td>7.</td>
<td>Flowchart of Main Loop Over Reactions in Subroutine C@MM</td>
<td>50</td>
</tr>
<tr>
<td>8.</td>
<td>Flowchart of the Solution for the Derivatives of the Flow Variables in the Numerical Integration</td>
<td>75</td>
</tr>
<tr>
<td>9.</td>
<td>Flowchart for the Determination of Sonic Conditions</td>
<td>79</td>
</tr>
<tr>
<td>10.</td>
<td>Three Cases in the Algorithm for Finding the Throat Temperature</td>
<td>81</td>
</tr>
<tr>
<td>11.</td>
<td>Flowchart of Subroutine FRøZEQ</td>
<td>83</td>
</tr>
<tr>
<td>12.</td>
<td>Flowchart of Subroutine GEøM</td>
<td>93</td>
</tr>
<tr>
<td>13.</td>
<td>Approximations to the Density-Area Ratio Relation</td>
<td>95</td>
</tr>
<tr>
<td>14.</td>
<td>Overall Flowchart of Subroutine MØDEL</td>
<td>110</td>
</tr>
<tr>
<td>15.</td>
<td>Flowchart of the Normal Shock Solution</td>
<td>113</td>
</tr>
<tr>
<td>16.</td>
<td>Flowchart of the Solution for Conditions at the Inviscid Stagnation Point</td>
<td>116</td>
</tr>
<tr>
<td>17.</td>
<td>Flowchart of Subroutine NØNEQ</td>
<td>122</td>
</tr>
</tbody>
</table>
LIST OF ILLUSTRATIONS
(Continued)

18. Flowchart of Initialization Section of Subroutine NØNEQ ................................. 124
19. Flowchart of Switch to Downstream Region in Subroutine NØNEQ ............................. 126
20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine NØNEQ ..................................................... 127
21. Flowchart of Perturbation Solution and Switch to Integration in Subroutine NØNEQ ................................. 129
22. Flowchart of Controls for the Nonequilibrium Integration in Subroutine NØNEQ ................................. 134
23. Overall Flowchart of Subroutine OUTF ........................................... 141
24. Flowchart of the Operations Produced by Calling Entry OUT2 of Subroutine OUTF ................................. 143
25. Iteration Scheme of Third Option for Specifying Reservoir Conditions ..................................................... 172
26. Flowchart of Subroutine RNKT ........................................... 175
27. Schematic Flowchart of Subroutine XSECT ........................................... 197
28. Geometry of the Condition ICØND = 1 ........................................... 578
29. Geometry of the Condition ICØND = 2 ........................................... 582
30. Geometry of the Condition ICØND = 3 ........................................... 584
31. Geometry of the Condition ICØND = 4 ........................................... 584
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, O-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 425000 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 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.

AESØLN
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 GEØM.

AGSØLN
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 FRØZEN, 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 NØNEQ.

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 FRØZEN, EQUIL, and BLCALL.

BLCALL
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.

**CİOMM** 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 $\text{NOE}_{\text{EQ}}$ and RNKT.

**DSMSØL** Performs simultaneous solution of a system of linear, inhomogeneous equations by calling subroutine SIMQ. Called by EQCALC, EXACT, NEWRAP, $\text{NOE}_{\text{EQ}}$, 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, $\text{NOE}_{\text{EQ}}$, 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 TRANSPE.
**ELTIME**  Prints elapsed time since the beginning of the run and time since the last previous time message. Called by MAIN and MØDEL.

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

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

**(EQUIL)**  Entry point in subroutine FRØZEQ; 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, CØMM, EQCALC, NEWRAP, ØUTL, PRØP, 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 AGSØLN, AXFIT, FRØZEQ, MAIN, and NEXTMP.

**(FINDXC)**  Entry 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.

**(FRØZEN)**  Entry point in subroutine FRØZEQ; 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.

**GEOMAR** 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 THROAT.

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

**GMAR2** Entry point of subroutine GEOMAR; 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 GEOMAR; computes the profile ordinates, $y, z$ and their derivatives $dy/dx, dz/dx$ at a specified axial coordinate $x$ in a channel. Called by AESOLN, 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 TRANSP.

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

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

MATINV Computes the inverse $\alpha_i^j$ of the matrix $\underline{\alpha}_i^j$ specifying the elemental composition of the independent species. Called by INGAS.

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

NEWRAP Computes the equilibrium flow conditions at specified temperature and entropy. Called by FRGZEQ, MAIN, NSEQ, and NRMAX.

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

NSEQ Organizes the nonequilibrium flow solution. Called by MAIN.

NRMAX 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.
ØUT1  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 ØUT1; 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 Te. 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 Qc 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 $\varphi = Ar^{-\eta}$. 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 $I(\ )$.  

---

-10-
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.

RESTM P 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.

(STUNTS) 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, MØDEL, NEWRAP, and STUNTS.

(THERM) Entry point in subroutine THERM; computes only the species specific heats at a specified temperature. Called by MAIN, ØUT1, and TRANSP.

THROAT 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.

TRANS P Organizes the transport property calculations. Called by BLAYER, FRÔZEQ, MAIN, MÔDEL, ÔUT1, WEDGE, and STUNT2.

(TRANSX) Entry point in subroutine TRANS P; 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 MÔDEL.

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

XSECT Sets arrays required for applying the pre-coded 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 (SHÔCK, 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.FROZ Calculate equilibrium solution to throat, to obtain displacement thickness at throat.
S.INIT Recalculate reservoir conditions corrected for displacement thickness at throat.
S.FROZ 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 NO320.....

N MSG FILE REQ TAPE 01 FH432 0 FSTRN 00

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-PUTOIN-

QCQU-QEQP-QEX-QINTRP-QLJ-QMX-QREPP-QSAME-QTAB-FLCOND-KANDMU-

OUTI-MODEL-PLAYER-WEDGE-GEOMAR-RADIUS-SHOCK-DUMP-TERM-WESGLN-

NEXTMP-FINDX-A'OLN-NEWRAP-DSMOL-SI-MC-ELC-ELTIME

S.READ  SEG READ

S.LIST  SEG LIST

S.STUN  SEG STUNTS

S.DUMP  SEG DUMPEX

S.INIT  SEG INIT-RESTMP-INGAS-MATINV

S.XSEC  SEG XSECT-AXSECT-DXSECT-CXSECT

S.FROZ  SEG FROZEQ-PROP

S.NONE  SEG NONEQ-PRTA-BLCALL-THROAT-EXACT-FPART-DERIVS-PIOMEG-GEOM-AESOLN-

COMM-*(RNKT,S.N2)

S.N2  SEG AXFIT-PERT

S.OUT  SEG OUT

XQT CUR

TRW C

JT C

EF C

REL C

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

Z RUN NO320.....

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
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 "FØR" cards. The "RUN", "MSG", "ASG", "FØR", "MAP", "XQT", "TRW", "ØUT", "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 WESØIN. 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 produc-
tion 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 Div-
ision, an available large compiler (FORTHMAX) is used to pro-
cess 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 RESET and TIME, which are available in
the 1108 system library. In the IBM 360 version in opera-
tion at Avco Systems Division, RESET and TIME are simu-
lated using the system library routine ACUCPU, which gives
the unexpended CPU time remaining before automatic job ter-
mination at TIME.$\varnothing$.

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

(1) An IMPLICIT REAL$\times8$ (A-H, $\varnothing$-Z) card must be in-
serted into each NATA routine except ELTIME,
$\varnothing$UT, RESET, SHOCK, SIMQ, and WES$\varnothing$IN. In MAIN,
this must be the first card apart from comment
cards. In the other routines, the IMPLICIT
statement must directly follow the SUBROUTINE,
FUNCTION, or BLOCK DATA statement.

(2) The subroutines EXP and RESET must be inserted.

(3) If a special system library for inter-
ceting the
single precision Fortran function names as double
precision functions is not available at the com-
puter installation, Fortran subroutines to de-
fine the single precision names as double pre-
cision 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=VB, 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 ISW1A 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( ).
CALL EQUIL(1)
Compute ATH
SMAS=45.28+0.6N/A

CALL INIT
CALL INITA

ICASE=17

CALL OUT

Print reservoir conditions

Figura 5b. Flowchart of NATA Main Program (Part 6)
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 NTRAN is checked. NTRAN is a switch to allow the suppression of all transport property calculations throughout NATA. With NTRAN = .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 NTRAN = .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 /ERRPR/. This indicator is set to .FALSE. near the beginning of each case. If the programmed checks in any subroutine subsequently detect an error, ERR is set to .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 FRZEN, CALL EQUIL(2), and CALL NÖNEQ 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, NATÆ 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, NATÆ 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 NATÆ 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), NATÆ generates one or more flow solutions.
If ISW6A 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 SUPGØ = .TRUE., and TRANSX was called in a previous case if IRCØ = 0. If NØTRAN 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

\[ C_{PW} = \frac{R_0}{W_c} \sum_{k=1}^{n_c} X_k \frac{C_{pk}}{R_0} \tag{1} \]

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 FRSEQ 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 /CONVRT/. 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 \( \frac{T}{T_0} \), \( \frac{P}{P_0} \), etc., are now set to their reservoir values and subroutine \( \PhiOUT1 \) 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 = NPR0FL(1), the type of solution (ISPLN = 1 for frozen flow), and the reservoir conditions.

3.1.10 Frozen Solution

To generate the frozen flow solution, MAIN calls the entry FRZEN of subroutine FRZEQ. It then calls OUT1 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, OUT1 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 \( \alpha \) in the area-density relation \( I(383) \) assumed in the solution by the inverse method upstream of the throat. The parameter \( \alpha \) is determined by a Newton-Raphson solution of equation \( I(385) \) or \( I(386) \). The iteration is started using the initial estimate

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

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

\[
\ln (\alpha + 2) \approx 2 + \frac{\alpha}{2} - \frac{\alpha^2}{8}
\]

Then to obtain starting conditions for the perturbation solution, MAIN sets the nondimensional temperature \( CT \) to \( 1-DELT \), calls subroutine \textsc{newrap} to compute the equilibrium flow conditions at this temperature, computes the corresponding effective area ratio \( AFNTS \), and calls subroutine \textsc{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 \textsc{nnoneq} 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 \textsc{call dumpe}, 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 \( DATAPE = .TRUE. \), an end file mark is written on tape 8 and the tape is rewound.

3.2 Subroutine \textsc{_areas}

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 S1, the derivative of the geometric area ratio S2, and the square root of the geometric area ratio SQRTA. To set S1 and S2, a call to AESLOLN(X) must be preceded by a call to GE@MAR(X,S1,S2). In addition, if NPRFLS = 1 (the flow is in a nozzle, not a channel) and JDIM = 1 (the nozzle is axisymmetric), then SQRTA must be computed in the calling routine before AESLOLN 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 S1 of common block /AEGE@/, replacing the geometric area ratio.

In addition, the subroutine computes the derivative, \( \frac{dA_e}{dx} \), 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**

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

**Channel**

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

3.3 Subroutine AGSØLN

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 \( \delta^* \) are given. These input data are provided in the argument list: \( A_e = AE, \delta^*_i = DEL(i), \) where the index \( i \) runs from 1 to 2 in the case of a channel. The argument list also includes the indicator UPDØWN, 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 AGSØLN, a numerical Newton-Raphson method is used to determine \( A_g \). Equation I(135) is written as

\[
A_g = A_c + \frac{y \delta_2^* + z \delta_1^*}{Y_0 z_0}
\]  \hspace{1cm} (7)

where

\[
A_c = (1 - \frac{\delta_1^*}{Y_0}) (1 - \frac{\delta_2^*}{z_0}) A_e - \frac{\delta_1^* \delta_2^*}{Y_0 z_0}
\]  \hspace{1cm} (8)
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

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

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, $A_g$ is set equal to $A_e$, 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

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

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 $UPD\Down$. When such a condition is detected, AGSØLN 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 $A_e$ 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 curvfit.

### 3.4 Subroutine AXFIT

This subroutine solves for the value of $x$ corresponding to a given effective area ratio ($AFNTS$) during the initial (perturbation) portion of the nonequilibrium solution. If the solution is being calculated without the boundary layer ($ISW3B = 0$), then the geometric area ratio is equal to the effective area ratio, and $x$ (denoted by $CX$) is determined by calling subroutine FINDX. If the boundary layer is included in the solution, but subroutine THROAT has not yet been called to reset $IUPD$ 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 FINDX. However, if the boundary layer is included and $IUPD = 0$, then the inviscid flow is coupled to the boundary layer and...
x is accordingly determined by calling AGS\$LN. Both FIND\$ 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 TRANS\$. 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 TRANS\$. 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(l,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(l,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 LLth 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 DO 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 \( \tau_w \) and heat flux \( q_w \) are omitted.

\( IP\) is a counter for the flow points at which boundary layer calculations are done in each flow solution. \( IP\) is initialized to 1 in subroutine FRZEQ for frozen and equilibrium solutions and in subroutine NONEQ for nonequilibrium solutions. Each time BLAYER is called with FINAL = .TRUE., \( IP\) 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 (\( IP = 1 \)), 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 \( PR = Pr_w \) and \( VISC = \mu_w \) at the wall temperature. The wall is assumed to be catalytic; hence, these transport property calculations are based on the mole fractions \( QPJ(K) \) 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:

\[
\begin{align*}
\text{ITYPE} & = 1 \quad \text{Two-dimensional nozzle} \\
\text{ITYPE} & = 2 \quad \text{Axisymmetric nozzle} \\
\text{ITYPE} & = 3 \quad \text{Rectangular channel}
\end{align*}
\]

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 $r_{2}/y_{0}$. In the case of a rectangular channel, for each profile, $A_{1}$ 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 $\frac{dA_{g}}{dx}$. In general,

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

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

$$
\frac{dy}{dx} = y_{0} \frac{dA_{g}}{dx}
$$

and hence

$$
\cos b = \frac{1}{\sqrt{1 + y_{0}^{2} (\frac{dA_{g}}{dx})^{2}}}
$$

In the axisymmetric case, from I(116),

$$
\frac{dy}{dx} = \frac{y_{0}}{2A_{g}} \frac{dA_{g}}{dx}
$$

and thus

$$
\cos b = \frac{1}{\sqrt{1 + \frac{y_{0}^{2}}{4} (\frac{dA_{g}}{dx})^{2}}}
$$

The derivative $\frac{dA_{g}}{dx}$ in (16) and (18) is obtained by calling subroutine GE$\emptyset$MAR. In the case of a channel, $dy/dx$ is computed by calling the entry GM3 of GE$\emptyset$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 EK1) is calculated from I(201) at the first flow point (IP\( \phi \)INT = 1), and then held constant through the solution.

The integrand \( f \) of I(172) is represented by \( \varnothing \)RDIN(L). Actually, \( \varnothing \)RDIN(L) differs from equation I(171) by a constant factor, which cancels out in the calculation of \( n' \) 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\( \phi \)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

\[
M A_g = \text{constant} \tag{19}
\]

The constant in (19) is represented by CC. At each point in the integration from \( x_0 \) to \( x_1 \), the geometric area ratio \( A_g = AGL \) and the quantity \( AGL \) corresponding to \( r^2 \) are calculated by calling subroutine RADIUS, \( M \) is obtained from (19), and a quantity \( \varnothing \)RD1 proportional to \( \varnothing \)RDIN(L) is computed with the
approximation \( a_0/a_0 \approx p_0/p_0 \approx 1 \). 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, \( DLM(L) = dln M/dx \) 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 \( dln M_e/d\xi \) for the \( L^{th} \) profile is denoted by \( DLM(L) \). It is determined by first computing \( dln M_e/dx \) and then correcting for the inclination of the nozzle or channel surface to the axis. For each flow point beyond the first, \( dln M_e/dx \) is calculated from rate data if the flow solution is of non-equilibrium 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), \( dln M_e/dx \) is calculated by numerical differentiation. If the inviscid solution is being calculated by nonequilibrium integration, \( DLM(L) \) 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

\[
\frac{du}{dx} = - \frac{1}{u} \sum_{j=1}^{n} \left[ \gamma_j \frac{d\gamma_j}{dx} + H_j \frac{dH_j}{dx} \right]
\]

(21)
The variables used in the code are

\[ \text{SHJ}(j) = \frac{H_j}{R_j T_0} \quad (22a) \]

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

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

\[ \text{GJ}(j) = \gamma_j \quad (22d) \]

Hence (21) is coded as

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

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

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

\[ \text{DUDX} = \frac{\text{DCHA}}{\text{SU}} - \frac{\text{CMA}}{\text{SU}} \left[ \text{GJ}(1) \cdot \text{CCPJ}(1) \cdot \text{DTE} 
+ \text{SCPGH} \cdot \text{DT} + \text{SHJDGJ} \right] \quad (24) \]

where

\[ \text{DCHA} = \frac{W_0}{R_0 T_0} \frac{dh_0}{dx} \quad (25) \]

\[ \text{SCPGH} = \sum_{j=2}^{n} \text{GJ}(j) \cdot \text{CCPJ}(j) \quad (26) \]
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 IPØINT = 2, a simple first-order difference expression

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

(27)

is used, where M is the Mach number at the current point and \( M_P \) that at the previous point. For IPØINT \( \geq 3 \), 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

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

(28)

from which

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

(29)

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

\[
c_2 = \frac{M-M_p}{x-x_p} - \frac{M-M_{pp}}{x-x_{pp}} \quad (31)
\]

Now that \( c_2 \) is known, \( c_1 \) can be obtained from (30b);

\[
c_1 = \frac{M-M_p}{x-x_p} - c_2 (x + x_p) \quad (32)
\]

d\( M/dx \) is then calculated from (29), and \( DLM(L) \) is obtained from

\[
\frac{d \ln M}{dx} = \frac{1}{M} \frac{dM}{dx} \quad (33)
\]

3.6.5 Calculation of Boundary Layer Thicknesses

Next, the increase in \( \xi \) in the step from the previous flow point to the current one is calculated,

\[
\Delta \xi = \frac{\Delta x}{\cos b} \quad (34)
\]

and \( \xi \) is incremented by \( \Delta \xi \). 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 \( d \ln M/d\xi \) by

\[
\frac{d \ln M}{d\xi} = \cos b \frac{d \ln M}{dx} \quad (35)
\]

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

\[
\pi (R_0)^2 = \text{throat area} \quad (36)
\]
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 \( \overline{\eta} \), 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

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

(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 \( DEBL(L) \) 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 THRTAI.* The only effect of the coupling in subroutine BLAYER is that the derivatives DDEBL(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 DØ 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 QWDØT(L) from I(195). Then, if L is less than NPRFLS, control is transferred to the end of the DØ 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 DØ 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 PERTGJ(J) are added to the concentrations \( \gamma \) = GJ(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
Indeed, subroutine PRTA (which controls the output of the nonequilibrium flow solution) calls BLICALL 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/TRANS7, 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 \( C\Omega MM \)

The function of subroutine \( C\Omega MM \) 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.

\( C\Omega MM \) is called only from subroutine DERIVS, and the call to \( C\Omega MM \) 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.

\[
\begin{align*}
\text{ALGJ(J)} & \quad \ln \gamma_j \\
\text{BETA(I,J)} & \quad \beta_{ij} \\
\text{CHI(I)} & \quad \chi_i \\
\text{CLKF} & \quad \ln k_f \\
\text{CLKR} & \quad \ln k_r \\
\text{CLNPI(I)} & \quad \ln p_i \\
\text{CLNIMC(I)} & \quad \ln (1 - \chi_i) \\
\text{CLNTD(L)} & \quad \ln T \text{ for } L = 1; \ln T_e \text{ for } L = 2
\end{align*}
\]
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLTBF</td>
<td>$\ln (\rho \sum \gamma_j)$, sum over third bodies</td>
</tr>
<tr>
<td>CLZ2</td>
<td>$\ln (R_0 T)$ or $\ln (R_0 T_e)$, argument in cm$^3$ atm/mole</td>
</tr>
<tr>
<td>CLZ3</td>
<td>$\ln (\rho R_0 T)$ or $\ln (\rho R_0 T_e)$, argument in atm-g/mole</td>
</tr>
<tr>
<td>CTD(L)</td>
<td>$T/T_0$ for $L = 1$; $T_e/T_0$ for $L = 2$</td>
</tr>
<tr>
<td>EF</td>
<td>$\epsilon_{fi}$</td>
</tr>
<tr>
<td>ER</td>
<td>$\epsilon_{ri}$</td>
</tr>
<tr>
<td>GJ(J)</td>
<td>$\gamma_j$</td>
</tr>
<tr>
<td>PI(I)</td>
<td>$p_i$</td>
</tr>
<tr>
<td>PICH(I)</td>
<td>$p_i \chi_i$</td>
</tr>
<tr>
<td>QDPE</td>
<td>$\dot{q}_e$</td>
</tr>
<tr>
<td>QDPR</td>
<td>$\dot{q}_r$</td>
</tr>
<tr>
<td>QF</td>
<td>$q_{fi}$</td>
</tr>
<tr>
<td>QR</td>
<td>$q_{ri}$</td>
</tr>
<tr>
<td>RH$\rho$P</td>
<td>$\rho$</td>
</tr>
<tr>
<td>RHPL</td>
<td>$\ln \rho$</td>
</tr>
<tr>
<td>SCPGH</td>
<td>$\frac{1}{R_0} \sum_{j=2}^{n} \gamma_j c_{pj}$</td>
</tr>
<tr>
<td>SU</td>
<td>$u/u_s$</td>
</tr>
<tr>
<td>TEP</td>
<td>$T_e$</td>
</tr>
<tr>
<td>T6</td>
<td>$\sum_{n=1}^{n} \nu_{ij} \ln \gamma_j$ or $\sum_{j=1}^{n} \nu_{ij}' \ln \gamma_j$</td>
</tr>
</tbody>
</table>
\[ T = \sum_{j} \gamma_j \]  sum over third-body species

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

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

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

\( \text{XMJATD(J,L)} \) \( \mu_j^o(T')/R_0 T' \), where \( T' = T \) for \( L = 1 \), \( T' = T_e \) for \( L = 2 \)

\( \text{XNUIJ(I,J)} \) \( \nu_{ij} \)

\( \text{XNUIJP(I,J)} \) \( \nu_{ij} \)

\( \text{Z1} \) \( \ln(1/u) \), for \( u \) in cm/sec

Subroutine \text{CPMM} 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.

3.9.1 Initialization Section

This section extends from the subroutine entry point through the statement RCALC = .FALSE. (card CPMM 107). It sets up the ALGJ array, computes a number of sums which are required later in CPMM or in other parts of NATA, calculates the nondimensional flow velocity \( SU \) from the energy equation I(245), and calls subroutine GEJM 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 ISW5B in blank common. 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 CFDBM, EXACT, RNRT, or PRTA 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 CFDMM, the variable ICYCLE is then reset from its preset value of 90000000 to |ISW5B|. The counter ICOUNT is incremented by 1 every time CFDMM 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 CFDMM, the dumps are switched on and are produced in all of the nonequilibrium routines until CFDMM is entered the next time, when they are switched off again. The coding of these controls is in the first nine executable statements of CFDMM.

3.9.2 Main Loop Over the Reactions

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

1) First, it computes the quantities $p_i \chi_i$ 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 $\dot{q}_e$ and the radiative loss $\dot{q}_r$ due to the reactions.

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

$$\ln p_i = \ln (\lambda^2) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \sum_{j=1}^{n} \nu_{ij} \ln \gamma_j$$ (39)
Figure 7. Flow Chart of Main Loop Over Reactions in Subroutine CM3

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

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

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

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

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

This algorithm allows the computation of a finite $P_i \chi_i$ value in situations where $P_i$ is so small that an exponential underflow occurs in (41), but $\chi_i$ is negative and so large that $P_i \chi_i$ 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

$$P_i = \left[ \frac{\rho V_i}{u} k_{fi} \prod_{j=1}^{n} \gamma_{ij} \right] \cdot \rho \sum_{tb} \gamma_k \quad (44)$$
in which the sum on the right is over the third-body species specified. The corresponding logarithmic expression is (39) with a term \( \ln \left( \rho \sum_k \chi_k \right) \) 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 \((k)\), \( \chi_{ik} = \chi'_{ik} = 1 \). Thus, a factor \( \rho \chi_k \) 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 \( k_{f_i} \) can depend on either \( T \) or \( T_e \), or both, or neither. The form of temperature dependence of \( k_{f_i} \) is specified by the array KTF(I). If the indicator \( KF = KTF(I) \) is equal to 1, \( k_{f_i} \) 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 non-dimensional temperatures into a two-element array CTD(L), and placing the logarithms of the two temperatures (expressed in degrees Kelvin) into an array CLNTD(L). Both cases are then treated by a single formula involving CTD(KF) and CLNTD(KF), i.e., statement 190 of CGMM, which evaluates \( \ln k_{f_i} \) from

\[
\ln k_{f_i} = \ln C_i + \gamma_i \ln T_k - \frac{E_{f_i}}{R \theta_k}\]

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

\[
k_f = C_i T_e \left( 1 - e^{-E_i/R_0T} \right)
\]
where $E_i$ is a parameter with dimensions of energy. For $K_{TF}(I) = 4$,

$$k_f = \frac{c_i T_e \gamma_i}{\max (1, \tau)}$$  \hspace{1cm} (47)

where

$$\tau = b \rho \gamma_a R$$  \hspace{1cm} (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 $K_{TF}(I) = 5$,

$$k_f = \frac{c_i}{\sqrt{R}}$$  \hspace{1cm} (49)

In NATA, the reverse rate constant $k_{ri}$ is calculated from the detailed-balancing relation $I(277)$, using the equilibrium constant $K_i$, equation $I(278)$. In electronic nonequilibrium gas models, the coding of $C\tilde{O}MM$ allows for three types of temperature dependence of $K_i$. If the indicator $KR = K_{TF}(I)$ is zero, the reverse reaction is assumed to occur only at a negligible rate. If $KR$ is 1, $K_i$ is calculated from $I(278)$ using the gas temperature $T$. If $IR = 2$, 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 $L = KR$ so that both cases can be treated using a single formula, card $C\tilde{O}MM$ 153.

Cases in which one or more of the concentrations $\gamma_j = 0$ require some special consideration because of the appearance of $\rho \gamma_j$ in the formulas used in $C\tilde{O}MM$. 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 $G_J(J)$ could underflow as a result of its decrease during the flow solution. However, this is most unlikely to occur in practice because, to underflow, $G_J(J)$ must fall below $10^{-75}$ (on the IBM 360/75) or below $10^{-308}$ (on the UNIVAC 1108). Since $G_J(J)$ 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 $G_J(J)$ would be stopped, long before underflow was reached, by the freezing of minor species (Section 7.5.3 of Volume I).

(2) A zero $G_J(J)$ 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 COMM is designed to cope with zero concentrations if they should occur. If one of the species concentrations, $\gamma^i$, 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)
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 $\gamma_j$ 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 CIIH, 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.,

$$\exp(L + \text{other terms}) = 0 \quad \text{(exactly)}$$

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

$$L + \text{other terms} - L = \text{other terms}$$

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 (4b) 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@M, $L$ 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, $v_{ik}$ 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 $\ln P_i^k$ proceeds normally. However, $\beta_{ik} = v'_{ik}$ is positive, so that the right hand side of (40) contains a term $v'_{ik} L$ which is negative and large in magnitude. The resulting exponential underflow during the evaluation of (42) then gives $\mathcal{X} = 1$. 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

$$\ln P_i = \ln \left( \frac{1}{n} \right) + \ln k_{fi} + (v_{i1} - 1) \ln \rho + \sum_{j \neq k} v_{ij} \ln \gamma_j + \gamma_{ik} L$$

(52a)
Here the term $\nu_i L$ on the right in (52b) is positive and 
very large; thus, equation (42) is not used and the product 
$\Pi_i \mathcal{X}_i$ is evaluated using (43), which gives

$$
\ln (1 - \mathcal{X}_i) = \beta_i \ln (\rho R_0 T) + \sum_{j \neq k} \beta_{ij} \left[ \ln \nu_j + \frac{\mu_{i0}}{R_0 T} \right] - \nu_{ik} \left( L + \frac{\mu_{k0}}{R_0 T} \right) 
$$

(52b)

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

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

$$
\ln P_i = \ln \left( \frac{1}{u} \right) + \ln k_{fi} + (\nu_i - 1) \ln \rho + \sum_{j \neq k} \nu_{ij} \ln \nu_j 
$$

(54a)

$$
\ln (1 - \mathcal{X}_i) = \beta_i \ln (\rho R_0 T) + \sum_{j \neq k} \nu_{ij} \left[ \ln \nu_j + \frac{\mu_{i0}}{R_0 T} \right] + (\nu_{ik} - \nu_{ik}) \left[ L + \frac{\mu_{k0}}{R_0 T} \right] 
$$

(54b)
Addition of these two equations gives

\[ \ln[p_i(1 - \chi_i)] = \ln(\frac{1}{\beta}) + \frac{1}{n} k \beta_i \ln(\rho) + (\nu_i - 1)\ln(\rho R_0 T) \]

\[ + \sum_{j \neq k} \nu_{ij} \rho_j \bar{\chi}_j + \sum_{j \neq k} \beta_{ij} \frac{\mu_{ij}}{R_0 T} + \beta_{ik} \frac{\mu_{ik}}{R_0 T} \]

\[ + \nu'_{ik} L \]  \hspace{1cm} (55)

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

\[ p_i(1 - \chi_i) = p_i - p_i \chi_i = 0 \]  \hspace{1cm} (56)

Also, (54a) and (41) give \( p_i = 0 \); thus, \( p_i \chi_i = 0 \). This result is correct for Case 4, as may be seen by setting one of the \( \chi_k \) to zero in I(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 \( \ln(\rho) = L \) when \( \chi_k = 0 \) 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 \( p_i \chi_i \). 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

\[ (\Delta \hat{q}_e)_i = k_{fi} \epsilon_f \sum_{j=1}^{n} (\rho \chi_j) \nu_{ij} - k_{ri} \epsilon_r \sum_{j=1}^{n} (\rho \chi_j) \nu'^{ij} \]  \hspace{1cm} (57a)
(57b)

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

which are obtained by combining equations I(320) and I(321b,c).

In equations (57), \((\Delta \dot{q}_r)_i\) and \((\Delta \dot{q}_e)_i\) denote the contributions of the \(i\)th reaction to \(\dot{q}_e\) and \(\dot{q}_r\), respectively. The energy partition parameters \(\varepsilon_{fi}, \varepsilon_{ri}, q_{fi}, q_{ri}\) are obtained by calling subroutine EPART. Effects of third-body species, if any, are taken into account using the quantity \(CLTF = \ln (\rho \sum_j \gamma_j)\), already evaluated during the computation of \(P_i \chi_i\).

3.9.3 Final Section

After the main loop over the reactions has been completed, \(\text{CNMM}\) checks the value of INEQ. If this indicator is zero (i.e., if the perturbation technique is still being used to generate the flow solution), the RETURN is executed immediately. If INEQ is nonzero (i.e., if the numerical integration of the rate equations has been started), \(\text{CNMM}\) calculates the non-dimensional pressure, \(PRES = \rho/\rho_0\). If a chemical nonequilibrium model is being used, the pressure is computed from equation I(273) in the form

\[
P = \frac{\rho}{\rho_0} T \frac{W_0}{W}
\]

(58)

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

\[
P = \frac{\rho}{\rho_0} \frac{\gamma e T_e + \gamma h T}{T_0} \frac{W_0}{W}
\]

(59)

In either case, \(\rho/\rho_0\) is then divided by \(\text{ROBRA} = \rho_0/\rho_0\) 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, C@MM computes the contribution to the electron energy transfer \( \dot{q}_e \) from elastic collisions, using the formula

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

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

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

and for ionic species

\[
\nu_{ei} = \frac{8}{3} \sqrt{\frac{n_i}{m_e}} \frac{n_{ie}^4}{(2k T_e)^{3/2}} \ln \left( \frac{k^3 T_e^2}{\pi n_{ie} e^{6}} \right)
\]

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
- \( Q_{ej}^{(1,1)} \) 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 NATP, the cross section $\sigma_{q_j}^{(1,1)}$ is obtained by calling the function PIOMEG(TEP). The constant factors in (61) and (62) are represented by $V_j1$, $V_j2$, and $V_n$, and are preset in a data statement in $\text{C@FM}$. 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, $\text{C@FM}$, 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 $\text{C@FM}$. 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 \( \text{CTE} = \frac{T_e}{T_0} \), and calls THERM to compute the species properties at the electron temperature. The values \( \mu_i^{\circ}(T_e)/R_0T_e \) 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 0), 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 C$1RMM 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 DMSFJL 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

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

(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

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

(64)

The constants \(b_j\) are stored as

$$b_j = AA(j, i + 1)$$

(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 \(B\). DMSFJL 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 RNAMz 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(ERR) RETURN pass control up 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 became 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 used 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\(\text{END}\)

Subroutine ELC\(\text{END}\) 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 \(Q(l,i,j) = k^{-\frac{1}{2}}\Delta^1_l\) 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\(\varnothing\) 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\(\varnothing\), and then sets ET\(\varnothing\) 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.
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 ETØ 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

$\text{IT} = 1 \quad \varepsilon_f = -a R_0 T_e$

$q_f = \varepsilon_0 - \varepsilon_f$

$\varepsilon_r = q_r = 0$

$\text{IT} = 2 \quad \varepsilon_f = -\frac{3}{2} R_0 T_e$

$q_f = \varepsilon_0 - \varepsilon_f$ 

$\varepsilon_r = q_r = 0$
In these formulas, \( \epsilon = E_0 \) denotes the energy available for partitioning between the electrons and the radiative losses, \( R \) the universal gas constant (1.9872 cal/mole-deg), and \( T_e \) the electron temperature (\( ^\circ \)K).

### 3.18 Subroutine EQCALC

EQCALC computes the conditions of thermochemical equilibrium at specified temperature \( T (\text{\( ^\circ \)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:

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

In these formulas, \( \epsilon_f = \epsilon_r = \frac{3}{2} RT_e \) and \( q_f = q_r = 0 \).
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 ($I = 1$ 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 $\tilde{T}$. The mole fractions for the dependent species ($I = ISC + 1$ to ISS) are then estimated using equation I(224), which can be written in the logarithmic form

$$\ln X_i = CHII(i-c) + \sum_{j=1}^{c} \tilde{\nu}_{i-c,j} \ln X_j$$  \hspace{1cm} (66)$$

where

$$CHII(i-c) = -\frac{\mu_{i}^{0}}{R_0 T} + (\nu_{i-c}^{*} - 1) \ln p + \sum_{j=1}^{c} \tilde{\nu}_{i-c,j} \frac{\mu_{i}^{0}}{R_0 T}$$  \hspace{1cm} (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 elsewhere in EQCALC.

Once initial estimates of the \( X_i \) and \( X_j \) have thus been determined, EQCALC enters the Newton-Raphson iteration based on equations (277) to (233). The iterative loop runs from statement 70 down to (but not including) statement 220. The constant terms \( F_j \), equation (227), in the system of linear equations (233a) for the corrections \( h_n^r \) are loaded into the \( M1 \)th column of the matrix AA by the D0 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 \(-X_n^r(\partial F_r/\partial X_n)_n^r\), equation (233b), is loaded into the first ISC rows and columns of AA by the nest of D0 loops ending at statement 140. Subroutine DSMS9L is then called to solve the system of equations for the correction factors \( h_n^r \). Upon the return from DSMS9L, the solution values for these quantities are obtained from the \( M1 \)th column of the matrix AA, where \( M1 = ISC + 1 \).

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 (234):

\[
x_k^{r+1} = (1 + h_k^r) \cdot x_k^r
\]

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

\[
x_k^{r+1} = x_k^r/2
\]

These calculations are done in the D0 loop ending at statement 180.

The corrected mole fractions for the dependent species \( c \) calculated from the equilibrium relations (224), based on the corrected \( X_i \) for the independent species. Equations (66) and (67) are again used, in the D0 loop ending at statement 200.

The convergence test for the Newton-Raphson iteration is that the absolute values of all the \( h_k^r \) 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, $ZSEN = S / R_0$, is calculated using equation I(30b). Note that the specific entropy of the gas is equal to

$$s = \frac{(S - \frac{R_0}{W}) - R_0}{ZSEN \cdot \frac{CRA}{ZCM}}$$

(70)

in cal/gm°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

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

(71)

The unknowns $X_j$ are

$$X_j = \frac{d \bar{\nu}_j}{dx} \quad \text{for } j = 1 \text{ to } n$$

(72a)
\[ x_{n+1} = \frac{1}{T_0} \frac{d\bar{T}}{dx} \]  
\[ x_{n+2} = -\frac{d \ln \bar{\rho}}{dx} \]  

The system of equations is

\[ \sum_{j=1}^{n} \alpha_{ji} \frac{d \bar{\gamma}_j}{dx} = 0 \]  
(73a)

\[ \sum_{j=1}^{n} \frac{\bar{\nu}_{i-c}}{\bar{\gamma}_i} \frac{d \bar{\gamma}_j}{dx} - \frac{1}{\bar{\gamma}_i} \frac{d \bar{\nu}_i}{dx} + \frac{T_0}{\bar{T}} \left[ \frac{\bar{H}/R_0 T_0}{\bar{T}/T_0} - 1 - \sum_{j=1}^{n} \bar{\nu}_{i-c,j} \left( \frac{\bar{H}/\bar{T}_0}{\bar{T}/T_0} - 1 \right) \right] \frac{d (\bar{T}/T_0)}{dx} 
+ (\bar{\gamma}_{i-c} - 1) \cdot \frac{d \ln \bar{\rho}}{dx} = 0 \]  
(73b)

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

\[ \sum_{j=1}^{n} \frac{\bar{H}_i}{R_0 T_0} \frac{d \bar{\gamma}_j}{dx} + \left[ \sum_{j=1}^{n} \bar{\gamma}_j \frac{u_{pj}}{R_0} \right] \frac{d (\bar{T}/T_0)}{dx} 
- \frac{\bar{u}}{u_s^2} \cdot \frac{u_s^2}{R_0 T_0} \frac{d \ln \bar{T}}{dx} = \frac{\bar{u}^2}{u_s^2} \cdot \frac{u_s^2}{p_{j} T_0} \cdot \frac{d \ln A}{dx} \]  
(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 $j \rightarrow i$, $i \rightarrow j$; equation (73c) is I(299) applied to the equilibrium flow; and (73d) is I(293) applied to the equilibrium flow. Note that

$$\frac{\overline{H}_i}{R_0 T_0} = SHJ(i) \quad (74a)$$

$$\frac{\overline{T}}{T_0} = CT \quad (74b)$$

$$\frac{\overline{u}_s^2}{u_s} = SU2 \quad (74c)$$

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

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 D$\phi$L, 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$\phi$ loops ending at statement 120; they are denoted by DGJ(J).
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 $h_0$. 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 $\sum_j^P C_j \gamma_j$, but for NT = 1 the sum is over all species (see subroutine C3MM).

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}{dT_0} ), ( \frac{d \ln \tilde{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 \tilde{A}}{dx} ), ( \frac{d(T_0/T_e)}{dx} )</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{d(T_0/T_e)}{dx} )</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·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

\[
V = V_o + \frac{A - AR}{|DADX|} \tag{75}
\]

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 GEOMAR. The same Newton-Raphson formulas and convergence test are used as in the area ratio case.

3.22 Subroutine FRZEQ

FRZEQ 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 FRZEN. Entry EQUIL(IPASS) gives the equilibrium solution.

3.22. Nondimensionalization of the Flow Variables

Throughout subroutine FRZEQ, 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)
- $FLUX$: $\frac{\rho u}{\rho_0 u_s}$, nondimensional mass flux
PRES \( \frac{p}{p_0} \), nondimensional pressure

RH\( \phi \) \( \frac{\rho}{\rho_0} \), 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

\[
u_s = \sqrt{\frac{R_0 T_0}{W_0}} \quad (76)
\]

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 FR\( \phi \)ZEN is called, subroutine FR\( \phi \)ZEQ determines the sonic conditions in the frozen flow before starting the main frozen flow solution.* This calculation is done by the statements from entry FR\( \phi \)ZEN 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 be a maximum. This condition follows from the continuity equation I(243) and the fact that the flow area at the sonic point is determined as follows: the temperature is initialized to its reservoir value, and is then decremented repeatedly by \( \Delta T \), 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 PR\( \phi \). 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, NRM\( \phi \)AX.
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 $T_1 > T_2 > T_3$ and the corresponding mass flux values $F_1$, $F_2$, $F_3$. $T_1$ and $T_2$ are always chosen in such a way that $F_2 > F_1$; thus, $T_1$ 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 $T_1'$, $T_2'$, $T_3'$ depend upon the value of $F_3 - F_2$. There are three cases, which are illustrated schematically in figure 10. In case A ($F_3 > F_2$), the three mass flux values form a monotonic sequence $F_3 > F_2 > F_1$. 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, $T_1'$ is set equal to $T_2$, $T_2'$ to $T_3$, and $T_3'$ is obtained by applying the same decrement $T_1' - T_2'$ to $T_2'$. In case B ($F_3 = F_2$), the sonic point is known to lie between $T_2$ and $T_3$. In this case, $T_1'$ is set equal to $T_2$, the temperature decrement $DELT = \Delta T$ is cut in half, and the remaining two temperatures are set to $T_2' = T_1' - \Delta T$ and $T_3' = T_2' - \Delta T = T_3$. In case C ($F_3 < F_2$), the sonic point could lie between $T_1$ and $T_2$ or between $T_2$ and $T_3$. To ensure that $T_1'$ lies above the sonic temperature, it is set equal to $T_1$. The decrement $\Delta T$ is halved as in case B, and $T_2' = T_1' - \Delta T$, $T_3' = T_2' - \Delta T = T_2$. If the sonic point actually lies between $T_2$ and $T_3$, 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

\[
F(T) = A + B(T-T_2) + C(T-T_2)^2
\]  

(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

\[
F = \frac{F_3 - F_1}{2(\Delta T)} (T - T_2) + \frac{F_1 + F_3 - 2F_2}{2(\Delta T)^2} (T - T_2)^2 \tag{78}
\]

Setting \(dF/dT = 0\) then leads to the final result

\[
T_* = T_2 + \frac{(F_3 - F_1)(\Delta T)}{2(F_1 + F_3 - 2F_2)} \tag{79}
\]

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 \(FR\)\(_{Z\text{EQ}}\) 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 \(THERM\) and \(PR\)\(_{QP}\) in the case of frozen flow and subroutine \(NEWRAP\) 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 \(OUT2\) of subroutine \(OUT1\).

The actual complexity of \(FR\)\(_{Z\text{EQ}}\), 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 FR3SEQ (Part a)
Figure 11c. Flowchart of Subroutine PROSEQ (Part c)
IPASS = 0       Boundary layer is omitted.
IPASS ≠ 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.

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

MODLPT = .FALSE. Regular point of the flow solution.
MODLPT = .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, FRPSEQ 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 XØ - 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, ISSONIC 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 $A_g \leq A_e$ 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 (ISSONIC = 0, CT = CTSTAR, CX = 0.) in order to determine the quantity
\[ 1 - \delta^* = \phi \text{MDST} \] (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 AGSOLN. In statement 340, AGSOLN 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 AGSOLN is then called again, with the same AFNX and the new DELBL produced by the call to BLAYER. This second call to AGSOLN 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 WS\text{SAVE} 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 FROZEQ 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 $X_{MODEL}$ 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 $X_{MODEL}$. If not, the flow solution proceeds normally. If $CX \geq X_{MODEL}$, the iteration to determine the $CT$ value corresponding to $CX = X_{MODEL}$ 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 $X_{MODEL}$ is a numerical Newton-Raphson technique embodied in statement 490:

$$ T = T_1 + (T_2 - T_1) \cdot \frac{X_m - X_1}{X_2 - X_1} $$

(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 $X_{MODEL}$.

The convergence criterion for the iteration is that $CX$ differ from $X_{MODEL}$ 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, M0DEL is reset to .FALSE., and subroutine M0DEL is called to perform the required calculations of conditions on models. Also, the indicator SKIP is set to .TRUE., and the next XMODEL is determined by calling NEXTMP. Before resuming the main flow solution, FRZEO checks whether the next XMODEL 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 XMODEL.

Once all of the XMODEL 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, XMODEL 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 M0DEL 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 GEOM

The function of GEOM 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 THRØAT, 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 THRØAT, the density is obtained from the analytical density-area relation (81):

\[ \left( \frac{\rho}{\rho_0} A_e \right)^2 \left[ 1 - \left( \frac{\rho}{\rho_0} \right)^\alpha \right] = C \]  

and the effective area ratio is then obtained from the continuity equation I(266). The constants \( \alpha \) and \( C \) in (81) are determined in the main program from sonic-point conditions, prior to the first call to GEØM, as explained in Section 7.4 of Volume I and in Section 3.1 of the present report.

The overall logical structure of GEØM is diagrammed in figure 12. Immediately upon entry, the subroutine calls GEØMAR to compute the geometric area ratio \( S_1 \) and its derivative \( S_2 \) at the current position \( CX \). If the nonequilibrium flow solution being computed includes the boundary layer (ISW3B \( \neq 0 \)), then AESØLN is called to convert \( S_1 \) into the effective area ratio and \( S_2 \) into the derivative of the effective area ratio, based on the current displacement thickness.

If the flow solution is beyond the call to THRØAT (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 RHØ is calculated from AFNX using the continuity equation. Also, the derivative \( DLØGA = d/n A_e/dx \) is computed from \( S_1 \) and \( S_2 \).

On the other hand, if the numerical integration technique 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 GEOM
density $\rho_0$ is calculated from S1 by solving the density-area ratio relation (81), and the effective area ratio $AFNX$ is recomputed from $\rho_0$ using the continuity equation. The derivative $DL\Theta GR = d\ln (\rho / \rho_0) / dx$ is then computed for use in the rate equations (see the discussion of subroutine EXACT above).

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

$$A_e = \frac{\rho_0}{\rho} \sqrt{c \frac{1}{\alpha} \left(1 - \left(\frac{\rho}{\rho_0}\right)^\alpha\right)} \quad (82)$$

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 GEOM 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:

$$A_p = \frac{\rho_0}{\rho_p} \sqrt{c \frac{1}{\alpha} \left(1 - \left(\frac{\rho_p}{\rho_0}\right)^\alpha\right)} \quad (83a)$$

$$\rho_0 \left(\frac{dA}{d\rho}\right)_p = \rho_0^A_p \frac{\rho_p^A}{\rho_p} \frac{(1 + \frac{1}{2c}) \left(\frac{\rho_p}{\rho_0}\right)^\alpha - 1}{1 - \left(\frac{\rho_p}{\rho_0}\right)^\alpha} \quad (83b)$$

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

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, GEFIX uses the following algorithm:

1. If \( A_e \leq 1 \), \( \rho = \rho_\star \) (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:

\[
A_e = 1 + \frac{1}{2} \left( \frac{d^2 A_e}{d\rho^2} \right)_\star (\rho - \rho_\star)^2 + \frac{1}{6} \left( \frac{d^3 A_e}{d\rho^3} \right)_\star (\rho - \rho_\star)^3
\]

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

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

\[
\frac{\rho_0 - \rho_\star}{\rho_0} = S \sqrt{\frac{2c}{\alpha}} (A_e - 1) \equiv S(\delta\rho)
\]

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
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, pmax 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 \( \rho_p \) lies too far from \( \rho * \). Thus, if the first estimate \( \rho_p \) 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

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

from which

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

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

\[
A_e \approx \sqrt{c/(\rho / \rho_0)}
\]

from which

\[
\frac{\rho}{\rho_0} \approx \sqrt{c/A_e}
\]

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

\[
\frac{\rho_p}{\rho_0} = \min \left[ \frac{\rho_0}{\rho_0} + \frac{2c(A_e - 1)}{\alpha} \sqrt{\frac{c(A_e - 1)}{\alpha}} \right] \]

\[
\left( 1 - \frac{c}{A_e^2} \right) \right)^{1/\alpha}
\]

(89)
In the downstream region \( \frac{dA_e}{dx} > 0 \), it is

\[
\frac{\rho_p}{\rho_0} = \max \left[ \frac{\rho_\star}{\rho_0}, \sqrt{\frac{2C(A_e - 1)}{\alpha}}, \frac{\sqrt{C}}{A_e} \right]
\]  \hspace{1cm} (90)

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

\[
\frac{d \ln \rho}{dx} = \frac{2C}{\alpha \left( \frac{\rho A_e}{\rho_0} \right)^2 - (\alpha + 2)C} \cdot \frac{d \ln A_e}{dx}
\]  \hspace{1cm} (91)

which can be obtained by differentiating (82). The factor \( d \ln A_e / dx \) on the right is evaluated as \( S_2/S_1 \), 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'Hospital's rule:

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

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

\[
\left( \frac{d \ln \rho}{dx} \right)_\star = -\sqrt{\frac{C(d^2 \ln A_e / dx^2)_\star}{\alpha \rho_\star / \rho_0}}
\]  \hspace{1cm} (92)

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

\[
\left( \frac{d^2 \ln A_e}{dx^2} \right)_\star = \left( \frac{d^2 A_e}{dx^2} \right)_\star
\]  \hspace{1cm} (93)
by virtue of the conditions $A_{c*} = 1$, $(dA_{c}/dx)_{*} = 0$. To proceed further, it is necessary to consider the three geometry options separately:

(1) Two-dimensional nozzle

In this case, from equation I(126) with $y_{*}$ replaced by $y_{0}$ as usual,

$$\frac{d^{2}A_{e}}{dx^{2}} = \frac{1}{y_{0}} \left( \frac{d^{2}y}{dx^{2}} \right) - \frac{1}{y_{0}} \frac{d^{2}x}{dx^{2}} \tag{94}$$

The term $(d^{2} y_{*}/dx^{2})_{*}$ 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 I(122) (shape 2, circular arc convex toward the axis), the remaining term in the numerator of (94) is given by

$$\frac{1}{y_{0}} \left( \frac{d^{2}y}{dx^{2}} \right) = \frac{1}{y_{0}} \frac{d^{2}y}{dx^{2}} = \frac{1}{y_{0}P_{3}} \tag{95}$$

The second equality is obtained by noting that $P_{2} = 0$ for a throat section. Thus,

$$\frac{d^{2}A_{e}}{dx^{2}} = \frac{1}{y_{0}P_{3}^{2}(1 - \delta^{*}/y_{0})} \tag{95}$$

(2) Axisymmetric nozzle

From equation I(131) since $(dA_{c}/dx)_{*} = 0$ and $A_{c*} = 1$,

$$\frac{d^{2}A_{e}}{dx^{2}} = \frac{1}{1 - \delta^{*}/y_{0}} \left[ \frac{1}{A_{g*}} \left( \frac{d^{2}A_{g}}{dx^{2}} \right) \right] - \frac{1}{2A_{g*}^{3/2}} \left( \frac{dA_{g}}{dx} \right) \frac{2}{y_{0}} \left( \frac{d^{2}x}{dx^{2}} \right)_{*} \tag{96}$$
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,

$$\frac{d^2A_e}{dx^2} * \simeq \frac{(d^2A_g/dx^2)_*}{\sqrt{A_g^*(1 - \delta^*/Y_0)}} \tag{97}$$

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

$$\frac{d^2A_e}{dx^2} * \simeq \frac{2}{y_0^* P_3 (1 - \delta^*/Y_0) \sqrt{A_g^*}} \tag{98}$$

(3) Rectangular channel

From equation $I(134)$,

$$\frac{d^2A_e}{dx^2} * = \frac{1}{D} \left[ (y - \delta^*_1) \frac{d^2z}{dx^2} - \frac{d^2 \delta^*_1}{dx^2} + (z - \delta^*_2) \frac{d^2y}{dx^2} - \frac{d^2 \delta^*_2}{dx^2} \right] + 2 \left( \frac{dy}{dx} - \frac{d \delta^*_1}{dx} \right) (\frac{dz}{dx} - \frac{d \delta^*_2}{dx}) \tag{99}$$

where

$$D = (y_0 - \delta^*_{1*}) (z_0 - \delta^*_{2*}) \tag{100}$$

At the sonic point, the quantities $(\frac{dy}{dx} - \frac{d \delta^*}{dx})$ and $(\frac{dz}{dx} - \frac{d \delta^*}{dx})$ 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,
Evaluation of \((d^2z/dx^2)_*\) and \((d^2y/dx^2)_*\) from \(I(122)\) gives, finally,

\[
\frac{d^2A}{dx^2} \approx \frac{(y_0 - \delta_{1*}^*)(d^2z/dx^2)_* + (z_0 - \delta_{2*}^*)(d^2y/dx^2)_*}{(y_0 - \delta_{1*}^*)(z_0 - \delta_{2*}^*)} \tag{101}
\]

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

3.24 Subroutine GEÔMAR

GEÔMAR 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:

\[
\begin{align*}
\text{GEOMAR}(X, \text{ARATI}_0, \text{DERIVA}) & \quad (\text{IENTRY} = 1) \\
\text{GMAR}(X, Y) & \quad (\text{IENTRY} = 2) \\
\text{GMAR2}(X, Y, Z) & \quad (\text{IENTRY} = 3) \\
\text{GMAR3}(X, DYDX, DZDX, Y, Z) & \quad (\text{IENTRY} = 4)
\end{align*}
\]

The arguments have the following meanings:

\[
\begin{align*}
X & = \text{axial coordinate in nozzle (cm)} \\
\text{ARATI}_0 & = \text{geometric area ratio} \\
\text{DERIVA} & = \text{derivative of geometric area ratio (cm}^{-1}) \\
Y & = \text{profile ordinate (cm)}
\end{align*}
\]
2 = ordinate for second profile in a channel (cm)

DYDX = derivative of profile ordinate

DZDX = derivative of ordinate for second profile in a channel

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

\[
\frac{dy}{dx} = p_2
\]  

(103)

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

\[
\frac{dy}{dx} = \frac{x - p_2}{\sqrt{p_3^2 - (x - p_2)^2}}
\]  

(104)

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

\[
\frac{dy}{dx} = -\frac{x - p_2}{\sqrt{p_3^2 - (x - p_2)^2}}
\]  

(105)

The subroutine then calculates the geometric area ratio, if required, from equation I(119) 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.,

\[
\frac{\partial A_y}{dx} = \frac{1}{y_0} \frac{dy}{dx}
\]  

(2D nozzle)  

(106)
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 \( \sqrt{RSA} \). 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:

\[
\begin{align*}
A_ki &\quad AIN(K,I) \\
\bar{V}_{-c,j} &\quad CDIJ(I,J) \quad (I = i - c) \\
-1 + V^*_{-c} &\quad BET(I) \quad (I = i - c) \\
q_i &\quad QM(I) \\
W_i &\quad CGI(I)
\end{align*}
\]

In the subroutine, the array AIN is first initialized to the square submatrix \( \alpha_{-c,j} \) 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 I(6), and is stored in CDIJ. Then the singly subscripted array \(-1 + \nu_{i-c}^*\), where \(\nu^*\) is defined by equation I(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 I(9) and I(10); it is denoted by QM(I). Finally, the species molecular weights \(W_i = CGI(I)\) are computed as follows:

\[
C_i = \sum_{j=1}^{c} \alpha_{ij} A_j
\]  
\((109)\)

where \(A_j\) is the atomic weight of the \(j\)th element;

\[
W_i = C_i \quad (i = 1, \ldots, c) \quad (110a)
\]

\[
W_i = \sum_{k=1}^{c} \bar{\nu}_{i-c,k} W_k \quad (i = c + 1, \ldots, n) \quad (110b)
\]

For the rationale of equation (110b), see equation I(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 \((25/8)\sqrt{T} \ A_i(\mu)\) defined by equation I(87), for all species \(i\) in the mixture. The value of \((25/8)\sqrt{T} \ A_i(\mu)\) is obtained simply by dividing the factor \(ZM2(I)\) 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

\[
A(1) = (25/8)\sqrt{T} \ \bar{a}(K)
\]

and

\[
A(2) = (25/8)\sqrt{T} \ \bar{a}(\mu)
\]  

from equation I(86), using the values of the matrix elements \(a_{ij}(K)\) and \(a_{ij}(\mu)\) 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 $\Delta_{ij}^{(\kappa)}$ 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 $ZKINT \equiv K_{\text{int}} / \sqrt{T}$ in units of milliwatts/cm$^2$K$^{3/2}$, where $K_{\text{int}}$ 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. Subroutine 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 output for a previous case. If $\text{SUPG} = .\text{TRUE.}$, the gas model is the same as in the preceding case, and the description of the gas model is skipped. If $\text{SUPG} = .\text{FALSE.}$, the following 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 switching from the perturbation technique to numerical 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 $a_{ij} = LPIJ(I,J)$ 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 $\mu_{ij}$ 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 $A_{ij}^{-1}$ of the matrix $\gamma_{ij}$; see equation I(3) and associated discussion. MATINV computes the inverse of matrix $\gamma$ using the Gauss-Jordan reduction with the maximum pivot strategy (ref. 8).

3.31 Subroutine MODEL

Subroutine MODEL is the principal routine for calculations of conditions on models in the NATA code. MODEL 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, \textsl{MODEL} calls subroutine \textsl{WEDGE}. The overall structure of \textsl{MODEL} is diagrammed in figure 14.

3.31.1 General Description

Immediately upon entry, \textsl{MODEL} performs some checks to determine whether model condition calculations should be done. Such calculations are skipped if both of the logical variables \textsl{AXISYM} and \textsl{WEDGEM} are false, or if the free-stream 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 \textsl{ELTIME} is called with the argument \textsl{IP} = 0 to record the execution time at the beginning of the model calculations. \textsl{ELTIME} is called again, just prior to the \textsl{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, \textsl{MODEL} sets a number of flow quantities which depend upon the free-stream flow conditions and are thus constant for a given call to \textsl{MODEL}. When \textsl{MODEL} is first called in a given flow solution, the equilibrium and frozen shock density ratios \textsl{EPSILON} and \textsl{EPSF} have the value zero, set in the calling routine (\textsl{FRZSEQ} or \textsl{NONEQ}). These ratios are reset to the value 0.01. When \textsl{MODEL} is called with \textsl{EPSILON} = 0, it computes the Prandtl number at the model surface temperature, \textsl{PRW}, based on the cold species mole fractions \textsl{QPJ(K)}. This value of \textsl{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 \textsl{MODEL} are contained within a large D\$ circle with the D\$ index IS\$LN, 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 \((R_p/u_1)(du_\infty/dx)_s\), 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 \(h_d\) 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 \(SHJSA(I)\) 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$^3$/cm$^2$ to ft$^3$/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 SC@UT, 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 ITP@UT.

### 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 M@DEL is used as a first estimate; this value is stored as $T_2E$ (equilibrium shock) or $T_2F$ (frozen shock) in common block /STAG/. Prior to the first call to M@DEL, these values are initialized to the reservoir temperature.

The criterion for selection of $T_2$ is 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 + ρ₁u₁²(1 - EPS)
ICOUNT = ICOUNT + 1
ICOUNT > 30?
yes
no

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

EPS - EPSOLD ≤ -0.001?
yes
no

Recalculate P2

Frozen Shock
ICOUNT = ICOUNT + 1
ICOUNT > 30?
yes
no

CT = T2/CTAP
CALL THERM
Compute ZCH, P2, R2, EPS

H2 = HS - HD · EPS²

|ZCH -1| ≤ 0.001?
yes
no

EXIT from Normal Shock Solution

ERRL0 = ERZ
TL0 = T2
L0 = .TRUE.

ERZ = ZCH - H2
ERZ ≤ 0?
yes
no

Compute T2 by Interpolation
T = 1.1 T2
T2 = 0.9 T2

Flowchart of the Normal Shock Solution
be equal to the value \( H_2 \) obtained from the Rankine-Hugoniot energy equation for the shock, equation I(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 I(437), and the shock density ratio from I(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 I(438). In this case, the static enthalpy \( Z_{CH} \) and density \( Z_{RH} \) 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:

\[
(T_2)_{new} = (T_2)_{old} - \frac{ERZ \cdot (THI - TL)}{ERRHI - ERRL}
\]  

Here \((T_2)_{new}\) is the improved estimate resulting from the linear interpolation, \((T_2)_{old}\) is the previous estimate, \(ERZ\) is the enthalpy error obtained at \((T_2)_{old}\), \(ERRHI\) is the last positive error: \(ERZ\) obtained and \(THI\) the corresponding \(T_2\) estimate, and \(ERRL\) is the last negative error obtained with \(TL\) 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 \(L\) and \(HI\) which are initialized to "FALSE". \(L\) 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", \(T_2\) is increased by 10 percent. If \(ERZ\) is positive and \(L\) is "false", \(T_2\) is
decreased by 10 percent. The interpolation formula (114) is not used until both IØ and HI are .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 TS is taken to be the temperature T2 just behind the normal shock. Since T2 is known to be lower than TS, the logical variable IØ is not needed in the stagnation-condition solution. Linear interpolation is begun as soon as a positive ERZ is obtained, where ERZ is the difference between the known stagnation enthalpy HS and the local enthalpy at the stagnation point (ZCH) computed from TS and PS.

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 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. ISMCNR denotes the number of dependent species. Initially, this is equal to n-c=ISS-ISC.
Figure 16. Flowchart of the Solution for Conditions at the Inviscid Stagnation Point
The array \( CII \) is redefined as the logarithm of the equilibrium constants \( K \) for the dependent species:

\[
CII(i-c) = -\frac{\mu_i^0}{R_0 T} + \sum_{j=i_m}^{c} \beta_{i-c,j} \frac{\mu_j^0}{R_0 T}
\]  

(115)

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

\[
PGJ(i) = \frac{S_i^0}{R_0} - CRRB \cdot W_i
\]  

(116)

where \( S_i^0 \) 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*

\[
CRRB = \frac{S_0}{R_0 W_0}
\]  

(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 \( CII \) and \( PGJ \), defined above, and obtains a set of initial estimates of the species mole fractions \( X_i = \text{CAPX}(i) \) from equation (11):

\[
\text{CAPX}(i) = W \gamma_i
\]  

(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 $M2^{th}$ 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 $D\phi$ 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 "$D\phi$ 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 $D\phi$ loops ending with statement 150. The coefficients I(261b) for $j = c + 1$ and $k = i_m$ to $c$ are computed in the $D\phi$ loops ending at statement 170. The coefficient I(261c) for $j = c + 1$ and $k = c + 1$ is set up by the loop $D\phi$ 180. Finally, the coefficients I(261a) for $j = i_m$ to $c$ and $k = c + 1$ are loaded into $AA$ by the $D\phi$ loops ending at statement 210.

If the number of dependent species ISMCNR is zero a logical indicator $N_DEPS$ is set to .TRUE. at the beginning of NEWRAP. Tests on $N_DEPS$ then cause the subroutine to skip all of the $D\phi$ loops with $D\phi$ 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(I) 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 \(1 + h_n^F\) 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 \(X_i\), the other quantities in the calculation (\(ZP = \ln p/p_0\) and \(CGM(U(K) = \ln X_K)\), 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 $f_n K_d$ is denoted in NEWRAP by CHII(i), equation (115). The $f_n X_i$ 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 $h_n^r$ 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 XMODI 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 XMOD2 at which the width of the wide (MBLth) profile has the specified value TCAR(ITS), and sets XMODEL equal to XMOD2.

In the case of nozzle flow (NPRFLS = 1), the next model point may be either the next point (XMOD1) 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 XMØDPT, XMSET, and subroutine NEXTMP pertain to the calculation of conditions at model points, and are discussed below.
Entry

CALL DERIVS

Initializations

CALL NEXTMP

MDFLT? yes

CALL NEXTMP

MDFLT? no

Figure 17. Flowchart of Subroutine MNEQ

ORIGINAL PAGE IS OF POOR QUALITY
The remaining parts of this section discuss individual portions and aspects of subroutine NØNEQ in detail.

3.34.1 Initializations

Figure 18 is a flowchart of the initializations section of NØNEQ, 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 $\Delta T/T_0 = \text{DELT}1$, where the variable $\text{DELT}1$ is under input control and has a preset value of 0.01. This preset value of $\text{DELT}1$ gives fairly large steps in the perturbation solution. In cases including the boundary layer (ISW3B $\neq$ 0), to allow more accurate evaluation of the boundary layer integral $I(172)$, $\text{DELT}1$ 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 NØNEQ, 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 $\text{DELT}1$ in NØNEQ. 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 NØNEQ computes the rank $\text{IX}$ of $\beta(I,J)$ and executes an error exit if $\text{IX}$ 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 THRTAT. The conditions are as follows:

1. $DLOGA = d \ln \tilde{A} / dx$, where $\tilde{A}$ 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 $\tilde{A} = AFNX$ 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 THRTAT. In THRTAT, 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 $\delta MST(I)$ 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 NOHEQ is entered when a positive value of $d \ln \rho / dx = DLOGR$ 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 NONEQ
Figure 20. Flowchart of Restart of the Upstream Solution at the Switch Point in Subroutine N\_NEQ
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 $> ARBA = 5$, 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), PERTGJ(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 DELT is still equal to the reduced value of DELT1 (see part 3.34.1 above), DELT is reset to the input value of DELT1, which was saved in DELETSV. 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 HKXEQ
(3) The largest $\text{SDCHI}(I)$ value ($\text{DCHMAX}$) and the smallest ($\text{DCHMIN}$) are determined.

(4) The ratio $\text{RATI} \rho = \frac{\text{DCHMIN}}{\text{DCHMAX}}$ is examined. If this ratio is larger than or equal to a value $\text{DCHRAT}$ (preset in BLOCK DATA to $10^{-4}$), $\text{NONEQ}$ proceeds to the checks for the switch to numerical integration, item (5) below. If $\text{RATI} \rho$ is less than $\text{DCHRAT}$, 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 $\text{RATI} \rho$ is approximately independent of the size of $\text{DCHMAX}$; thus, the program responds to an excessively small value of $\text{RATI} \rho$ immediately, even though $\text{DCHMAX}$ may be much smaller or much larger than the value $\text{CCHI}$ at which the switch to integration is desired. To prevent the premature start of the numerical integration, $\text{NATA}$ artificially increases the rate constant for the reaction ($I = \text{IMAX}$) for which $|\text{SDCHI}(I)| = \text{DCHMAX}$, 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 $\text{RATI} \rho < \text{DCHRAT}$, $\text{NONEQ}$ determines whether this reaction involves a minor species (concentration less than or equal to $\text{GAMIN}$, preset to $10^{-10}$) whose concentration is being decreased by the reaction. If so, the rate constant for the $\text{IMAX}^{\text{th}}$ reaction is increased by a factor of $\text{RATI} \rho 2 = 1.1 \times \text{DCHRAT}/\text{RATI} \rho$, 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 $\text{DCHMAX}$ has reached the value $\text{CCHI}$ at which the switch to the numerical integration is desired. If $\text{DCHMAX}$ is less than $\text{CCHI}$, 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 MODLPT is set to \texttt{.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 MODLPT is \texttt{.TRUE.}, PRTA also calls subroutine MODEL 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 TSTP, 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 NEWRAP 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), N\texttt{NEQ} commences an iterative adjustment of the nondimensional temperature CT in order to obtain a flow point at which DCHMAX= \|\delta X 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, NOEQ 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 AXPIT 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).

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 PRHΦ are added to the corresponding equilibrium flow quantities CT, GJ(J), and RHΦ. 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 = |ΔX_i|_{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 NOEQ. 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 ≥ CXMAX or CT ≤ TSTØP, 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 + DELTAX, at the end of the forthcoming step is compared with XMØDEL and CXMAX. If CX + DELTAX is greater than either of these values, DELTAX is reduced to make the end of the step coincide with the lesser of
the two. If the end of the step will be at $X_{\text{MODEL}}$, then $\text{MODLPT}$ is set to .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 .TRUE. value of FAILED indicates that a validity check was not satisfied somewhere in RNKT, DERIVS, or C@MM. If FAILED = .TRUE., the step size is reduced by dividing DELTAX by SCD and the calculation is restarted at the beginning of the step, using the data saved in step (4). In the event of such a step restart, a counter ICOUNT is incremented. Also, $\text{MODLPT}$ is set to .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 .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(ICOUN). 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 NOREA C(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(ICOUN) is defined in Section 4.55.
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.

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 $0.7 \times \text{DXOLD}$. 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.

Control is now transferred to statement 210 ("MÅDLPT?" 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 NØNEQ. 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:

- $C_J$
- $D_C(I)$
- $G_S(Q(I))$
- $A_{IN}(J,K)$
- $D_C(A(K))$
- $A_{AA}(I, ISMCPL)$

\[
C_J \quad c_J \quad c_J'
\]
\[
D_C(I) \quad \delta c_J
\]
\[
G_S(Q(I)) \quad (\gamma_i')^2
\]
\[
A_{IN}(J,K) \quad A_{jk}
\]
\[
D_C(A(K)) \quad \frac{1}{(\gamma_k')^2} \sum_{j=1}^{C} \delta c_j \cdot A_{jk}
\]
\[
A_{AA}(I, ISMCPL) \quad b_i \text{ (before call to DSMSØL)}
\]
CDIJ(I,K) \( \vec{v}_{i-c,k} \)
AA(I,L) \( a_{i,l} \)
AA(I,ISMCPL) \( \delta \gamma_{c+1} \) (after call to DSMS\( \$ \)L)

The system of equations I(346) is solved for \( \delta \gamma_m \) by calling subroutine DSMS\( \$ \)L. After the adjusted concentrations of the dependent species \((i = c + 1 \text{ to } n)\) have been calculated from I(340) and those of the independent species \((i = 1 \text{ to } c)\) 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 XM\( \$ \)DEL. Whenever a particular model point has been reached and the model calculations have been done, the new XM\( \$ \)DEL value is obtained by calling subroutine NEXTMP(ITS,XM\( \$ \)DL,XM\( \$ \)DEL). The argument ITS is the index of the next specified test section diameter, and XM\( \$ \)DL 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 XM\( \$ \)DL, sets XM\( \$ \)DEL accordingly, and updates ITS or XM\( \$ \)DL as required. Before NEXTMP is called for the first time in N\( \$ \)NEQ, the initial values ITS = 1 and XM\( \$ \)DL = 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 XM\( \$ \)SET denotes that this initialization has already been performed.

The indicator M\( \$ \)DLPT, if .TRUE., denotes that the current flow point in the perturbation solution or the integration is a model point. If M\( \$ \)DLPT = .TRUE., P\( \$ \)TA calls entry P\( \$ \)UT2 of subroutine P\( \$ \)UT1 to print the flow conditions (even though the TPRINT criterion may not be satisfied) and calls M\( \$ \)DEL to compute and print the model test conditions.
When a model point is encountered during the numerical integration, \( \Delta X \) is reset to provide a flow point exactly at the model point. During the perturbation solution, \( \text{MODEL} \) is called at the first solution point beyond \( \text{XMODEL} \). To obtain a perturbation solution point coinciding with \( \text{XMODEL} \) 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 \( \text{NPNEQ} \) 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 \( \text{NPNEQ} \) 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 \( \text{FRZEQ} \). 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 \( \text{NEWRAP} \).

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 OUT1

Subroutine OUT1 is the principal output routine of NATP. 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 OUT2(ISFNL), 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 OUT2 causes a record to be written on the binary output tape (tape 8), for subsequent use by the NATP plot program. Figure 23 is an overall flowchart of the subroutine.

3.37.1 Entry OUT1

OUT1 is called (by MAIN) to print the conditions in the reservoir and at the frozen and equilibrium sonic points. Since a call to OUT1 precedes each of the three flow solutions, OUT1 is also used to initialize some quantities used by OUT2.

The chemically frozen specific heat ratio GAMMA is computed in a section of coding which is used by both OUT1 and OUT2. First, the entry THERML of subroutine THERM is called to compute the nondimensional species specific heats \( \frac{C_{pi}}{R_0} = \text{CCPJ}(I) \) at the current nondimensional temperature \( \frac{T}{T_0} \). The nondimensional molar specific heat of the gas mixture is then computed as

\[
CP \equiv \frac{C_P}{R_0} = \sum_{i=1}^{n} X_i \frac{C_{pi}}{R_0}
\]  

(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

\[
C_P - C_V = R_0
\]  

(121)

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

\[
\gamma = \frac{C_P}{C_V} = \frac{1}{1 - \frac{R_0}{C_P}}
\]  

(122)
ENTRY OUT1

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

CALL THERM1

Compute CP
Compute GAMMA

IFLAG?

1

Set output variables
POUT, UOUT, etc.

Compute FAREA
Compute TFLOW

Print Output

RETURN

2

Print flow conditions at a point in the solution

RETURN

ENTRY OUT2

IFLAG=2

Figure 23. Overall Flowchart of Subroutine 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 lbf/ft²sec.

Finally, the flow conditions, including the calculated mass flow TFLM, 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(ISWLN) 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 MUT2 of Subroutine MUT1 (Part a)
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 OUT2 is to re-express these quantities in the desired output units. The converted flow variables in output units are loaded into an array FVOUT(I), dimensioned (35). This scheme allows the use of compact WRITE statements based on implied DO loops. The converted quantities in FVOUT are also used in the calculations of conditions on wedge models; they are communicated to subroutine WEDGE through the common block OUTPUT. Definitions of the 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 FVOUT always have the meanings indicated:

(1) Axial coordinate, x (inches)

(2) Temperature, T (°K)

(3) Enthalpy, h (Btu/lb)

(4) Pressure, p (atm)

(5) Density, ρ (lbm/ft³)

(7) Velocity, u (ft/sec)

(8) Mach number, M

(9) Entropy, s (Btu/lbm−°R)

(10) Frozen specific heat ratio, γ
Effective area ratio, $A_e$

Reynolds number per foot

Molecular weight (g/mole)

Viscosity ($\text{lbm}/\text{ft-sec}$)

Electrical conductivity ($\text{mho/cm}$)

Element number 6 is absent from this list because its meaning varies with the type of case, as explained below. For ISW3B $\neq 0$, the Reynolds number per foot, viscosity, and electrical conductivity are calculated in BLAYER, and OUT2 obtains these data from common blocks /BLOUT/ and /TRPRP/. For ISW3B = 0, BLAYER is not called and these quantities must be computed in OUT2. The viscosity and conductivity are obtained by calling subroutine TRANSP.

If the boundary layer is included (ISW3B $\neq 0$), the ten elements FVOUT(16) to FVOUT(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 FVOUT(26) to FVOUT(35) contain the corresponding data for the second profile. The meanings for the elements numbered 17 to 25 are as follows:

Boundary layer displacement thickness (inch)

Boundary layer momentum thickness (inch)

Heat flux to nozzle wall (Btu/ft$^2$-sec)

Shear stress on nozzle wall (lbf/ft$^2$)

Recovery enthalpy (Btu/lbm)

Prandtl number at the reference temperature

Stanton number
(24) Reynolds number $Re_\theta$ based on momentum thickness

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

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

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

B. If ISW3B = 0, NPRFLS = 2: $F\psi U(6)$ 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 ≠ 0, NPRFLS = 1: $F\psi U(16) =$ geometric area ratio, $F\psi U(6) =$ nozzle diameter (inch).

D. If ISW3B ≠ 0, NPRFLS = 2: $F\psi U(6) =$ geometric area ratio; $F\psi U(16) =$ width of channel surface lying on the first profile (inch); and $F\psi U(26) =$ 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 IS\$L. In
the case of a nonequilibrium solution, the number of inte-
gration 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 init-
ialized in OUT1 to the seven-digit value 1000000, so that
when printed (with an I6 format) in the frozen and equi-
librium solutions, it gives a row of six asterisks.

Following the divider line, the first fifteen elements
of FV\$UT(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 FV\$UT(16) to FV\$UT(25)
or FV\$UT(35) are printed. For channel cases without the
boundary layer, the channel widths CHDIMS(I) are printed on
a separate line. If LIM\$UT is nonzero, the boundary layer
parameters stored in FV\$2(I,J) are printed on a separate
line. For nonequilibrium solutions based on an electronic
nonequilibrium model, the electron temperature, the radia-
tive and electronic energy transfer terms, and the total
enthalpy are loaded into TN\$UT(I) and printed.

Next, if ISW6B is nonzero and IS\$L 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 \$UT2 is called in the equilibrium and
nonequilibrium solutions, where IS6 = |ISW6B|. Finally, if
ISW6B is negative and IS\$L = 3 (nonequilibrium solution)
and if I6 = IS6, a table of reaction rate data is printed
out. This table includes Pi, \( \dot{\gamma}_i \), and \( P_i \dot{\gamma}_i \) for all of the
reactions \( i = 1 \) to \( r \), and \( \frac{d \dot{\gamma}_j}{dx} \) for all the species
\( j = 1 \) to \( n \).

If DATAPE is .TRUE., a record is written on tape 8 for
subsequent use by the NATA plot program.

3.38 Subroutine PERT

The function of subroutine PERT is to compute the per-
turbations in the flow variables by solving the following
system of \( n+2 \) linear equations:
Equation Index | Equation
---|---
1 to c | I(368) for \( j = 1 \) to c
\( c+1 \) to n | I(362) for \( j = c+1 \) to n
n+1 | I(374)
n+2 | I(380)

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 DSMSØL.

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
\( DØ \) loops ending at statement 20, the coefficients for equa-
tions 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
\( DØ 120 \) down to the second statement below 130 sets up the
coefficients for equation I(380) in AA(ISSP2,I). Finally,
the loop \( DØ 140 \) loads the constant terms of equation I(362)
into AA(I,ISSP3) for \( I = ISCPL \) to ISS. The remaining equa-
tions are all homogeneous, so that their constant terms have
been set by the initial zeroing of the AA array.

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

3.39 Function PIØMEG

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

3.40. Subroutine PR\( \beta \)P

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

The technique used by PR\( \beta \)P 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

\[
Z_P = \ln \left( \frac{P}{P_0} \right) = -\ln p_0 + \sum_{j=1}^{n} x_{j0} \frac{s_{j0}^o}{R_0} - \sum_{j=1}^{n} x_{j0} \gamma_{j0} s_{j0}^o - \frac{w_0 s_0}{R_0}
\]

(123)

The first sum on the right is evaluated in PR\( \beta \)P as

\[
\sum_{j=1}^{n} x_{j0} \frac{s_{j0}^o}{R_0} = w_0 \sum_{j=1}^{n} \gamma_{j0} s_{j0}^o = \text{CMA} - \text{S4}
\]

(124)

The remaining three terms on the right in (123) are computed as \( w_0 \cdot \text{ENT} \), where ENT has been calculated in PR\( \beta \)ZEQ before PR\( \beta \)P is first called:

\[
\text{ENT} = -\sum_{j=1}^{n} \gamma_{j0} \left( \ln \gamma_{j0} + \ln w_0 \right) - \frac{\ln p_0}{w_0} - \frac{s_0}{R_0}
\]

(125)

3.41 Subroutine PRTA

This subroutine controls the output of the flow conditions and executes the calls to the M\( \beta \)DEL routine during
the nonequilibrium solution. It also calls subroutine BICALL 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 BICALL 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 BICALL. For ISW3B = 0, the call to the boundary layer routine BLAYER is suppressed in BICALL.

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 PUT2 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 $\bar{\Omega}_{ij}$ 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 TRANSP/ (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 Qc, where Qc 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 Lth step of the computations (see Section 4.6 of Volume II), M is the location of the first parameter for the Lth 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 Lth 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 0.8Qc.

*Note that the number N of species in the cross section computations, which is stored in common block TRANSP/, is not necessarily the same as the number of species in the transport calculations which is stored in common block TRANSP/. 

-152-
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 $\bar{\Sigma} (1,1)$, $\bar{\Sigma} (2,2)$ and $B \bar{\Sigma} (1,1)$ computed for the step to subroutine PUTQIN in the three locations $\bar{\Omega}(1)$, $\bar{\Omega}(2)$, and $\bar{\Omega}(3)$ respectively of the dimensioned variable $\bar{\Omega}$. 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.

### TABLE II

**SUBROUTINES USED IN CROSS SECTION COMPUTATIONS**

<table>
<thead>
<tr>
<th>Option K$^{KQ}$</th>
<th>Subroutine Used</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>PUTQIN</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>QEXP</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>QEX</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>QTAP</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>QREPP</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>QLJ</td>
<td>1</td>
</tr>
<tr>
<td>9</td>
<td>QSAME</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>QMIX</td>
<td>2</td>
</tr>
<tr>
<td>11</td>
<td>Q11</td>
<td>2</td>
</tr>
<tr>
<td>12</td>
<td>Q12</td>
<td>1</td>
</tr>
<tr>
<td>13</td>
<td>Q13</td>
<td>2</td>
</tr>
<tr>
<td>14</td>
<td>Q14</td>
<td>2</td>
</tr>
</tbody>
</table>


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 $\Phi_M$ 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, so 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 ISW88th 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 $\gamma$, $Q$ 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 QCUL

Subroutine QCUL computes the factor $0.8Q_c$ in the Coulomb cross section from equation I(100b). Inputs for this computation are the temperature $T$ in $\text{K}$ and the
electron pressure \( X = n_e kT \) in atmospheres. The computed value of \( 0.8Q_c \) in \( 10^{-16} \) cm\(^2\) is returned in \( \Phi_M \). For \( C < 0 \), \( 4n(f \Lambda) \) 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 \( \overline{\Omega}(1,1) \) and \( B* \overline{\Omega}(1,1) \) in \( 10^{-16} \) cm\(^2\) 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

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

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

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

(127a)

where

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

and

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

(127b)

\( \gamma = 0.577216 \ldots \) 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 \( \overline{\Omega}(1,1) \) and \( B* \overline{\Omega}(1,1) \) in the subroutine. Since the cross section \( \overline{\Omega}(2,2) \) is not affected by exchange (ref. 10), subroutine QEX sets the contribution \( \Phi_M(2) \) to this cross section equal to zero.
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 \( 10^{-16} \text{ cm}^2 \) 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 \( \text{Å} \). These cross sections are obtained by linear interpolation in Monchick's tables (ref. 11) of \( \alpha \) versus \( I(1,1) \), \( A^* \), and \( B^* \), where

\[
\alpha = \ln \left( \frac{A}{kT} \right)
\]

\[
I(1,1) = \frac{\bar{\sigma}(1,1)}{(4 \pi \alpha^2 \rho^2)} \quad (128)
\]

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.

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 ØMEGAL, 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 ISWBB ≠ 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 A of the independent variable, and the two values of the variable from the TL array used in the extrapolation.

3.47 Subroutine QLJ

Subroutine QLJ determines the collision cross sections \( \Omega_{(1,1)} \), \( \Omega_{(2,2)} \), and \( B^* \Omega_{(1,1)} \) 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, OMEGAL, ASTAR, and BSTAR respectively in common block/TRANS4/, starting at location NL = 501 in the arrays.

3.48 Subroutine QMIX

Subroutine QMIX 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 IQ1 and IQ2 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 QMIX, these quantities are stored in the array

\[
SQT(K,I) = \frac{1}{2} \sqrt{Q(K, I, I)} \tag{129}
\]
Upon each entrance to the subroutine, the values of $SQT(1,I)$ 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 $SQT(1,I)$ and $SQT(1,J)$ to determine if they have been calculated previously in the loop, and, if not, computes the required values of $SQT$ from (129) for $K = 1, 2, \text{ and } 3$. The $SQT$ values for the pair are then used in equation I(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 $\varphi = Ar^{-\gamma}$, where $r$ is the distance between the species in $\hat{R}$. The cross section formulas for this potential are (ref. 13):

\begin{align*}
\bar{\Omega}(1,1) &= \Omega_{\text{MEGAL(ITL)}} / (1^{2/\gamma}) \\
\bar{\Omega}(2,2) &= A_{\text{STAR(ITL)}} \ast \bar{\Omega}(1,1) \\
B^*\bar{\Omega}(1,1) &= B_{\text{STAR(ITL)}} \ast \bar{\Omega}(1,1)
\end{align*}

(130)

where $\Omega_{\text{MEGAL(ITL)}}$, $A_{\text{STAR(ITL)}}$, and $B_{\text{STAR(ITL)}}$ 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 \(\Omega_{1,1}, \Omega_{2,2}, \) and \(B^* \Omega_{1,1}\).

3.51 Subroutine QTAB

Subroutine QTAB obtains the values of the cross sections \(\Omega_{1,1}, \Omega_{2,2}, \) and \(B^* \Omega_{1,1}\) for a pair of species at a given temperature \(T\) by linear interpolation in tables of \(\Omega_{1,1}, \Omega_{2,2}\) and \(B^* \) versus temperature for the species which are included in the input. The tabulated data are assumed to be given in the arrays \(\Omega\) MEGA1, \(\text{ASTAR}, \text{BSTAR}, \) and \(\text{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 \((108)\). 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 \(\Omega_{1,1}, \Omega_{2,2}, \) and \(B^* \Omega_{1,1}\) for a species pair from the generalized mixing rule given by equation \((109)\). As with the other type 1 subroutines discussed above, it returns the computed cross sections to subroutine PUTQIN in the dimensioned variable \(\varnothing M\).
3.54 Subroutine Q13

Subroutine Q13 computes the cross sections \( Q(K_1,I,J) \) according to equations I(110) and I(111) for a fixed value of \( K_1 \) specified in the cross section data \( (1 \leq K_1 \leq 3) \) and for a series of species pairs \( I,J \) given in positions \( LQ_1 \) through \( LQ_2 \) 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 \( \tilde{Q}(1,1) \), \( \tilde{Q}(2,2) \), and \( B \) \( \tilde{Q}(1,1) \) for a series of species pairs by multiplying the previously computed values of these cross sections by a constant factor, as indicated in equation T(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 \( I\text{TYPE} \), 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 \( I\text{TYPE} = 1 \) (two-dimensional nozzle), \( R \) is obtained by calling the entry GMAR of subroutine GEOMEMAR. Then \( AG = K/R0 \) where \( RO \) is the half-width of the throat gap, and \( AGJ = 1 \) since \( j = 0 \).

For \( I\text{TYPE} = 2 \) (axisymmetric nozzle), \( R \) is again obtained by calling GMAR, \( AG = (R/RO)^2 \), where \( RO \) here is the throat radius, and \( AGJ = AG \) since \( j = 1 \).

For \( I\text{TYPE} = 3 \) (rectangular channel), the ordinates \( YZ(1), YZ(2) \) of the two profiles are obtained by calling the entry GMAR2 of subroutine GEOMEMAR, \( AG = YZ(1) \cdot YZ(2)/Y0ZO \) where \( Y0ZO \) 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 \( L = 2 \).
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 ACM. 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 BLÖCK 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 M (denoted by FLÓW) 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

$$SMASS = \frac{\dot{M}}{D} \quad (131)$$

where $D$ is the throat gap. Since $SMASS$ is required in gm/cm$^2$-sec while $\dot{M}$ is in lb/in-sec and $D$ in inches, a numerical conversion factor from lb/in$^2$-sec to gm/cm$^2$-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 lb/sec, and

$$SMASS = \frac{4\dot{M}}{\pi D^2} \quad (132)$$

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

$$SMASS = \frac{\dot{M}}{D_1 D_2} \quad (133)$$

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°K. On the other hand, if the reservoir temperature is to be determined from mass flow data (ISW2B ≤ 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

\[ \text{HWINOV} = \left( \frac{136.7 \ p_0}{m_*} \right)^{2.52} \]  

(134)

where \( p_0 \) denotes the reservoir pressure in atmospheres and \( m_* \) the sonic mass flux in g/cm² sec. The numerical factor on the right in (134) differs from the coefficient (280) in Winovich's equation because it includes a factor converting \( m_* \) from g/cm²-sec to lb/ft²-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 BLCK 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 precoded data. For the Jth profile ($J = 1$ or 2), if NPRPL(J) is nonzero, the geometric data are looked up in the ZPRP array, which is set in BLCK DATA. This array occupies the common block /NOZZ/. If NPRPL(;) is zero, the geometric data are set on the basis of the geometric input variables such as NSECTS, DIAM, and DARAMI.
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 XM&DPT, a final point CXMAXI, and a number of points NM&DPT. From these data, subroutine READ computes a factor FACMP which is used to generate a geometric sequence of NM&DPT values of x extending from XM&DPT to CXMAXI. This factor is calculated from the formula

\[
FACMP = \left[ \frac{\text{CXMAXI}}{\text{XM&DPT}} \right]^{1/(\text{NM&DPT}-1)}
\]

### 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 IZER0. Subsequent calls to the entry TIME then give the elapsed time 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 \( g/cm^2/sec \) is given by SMAS 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 PRESA = \( p_0 \) 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 SMASS.

RESTMP starts the iteration with a standard guess \( T_0 = 10000^\circ\text{K} \). 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

\[
SM = \frac{\rho_\ast u_\ast}{\rho_0 u_s} = \frac{m_\ast}{\rho_0 \sqrt{R_0 T_0}/\gamma_0}
\]

From the perfect gas law,

\[
P_0 = \frac{\rho_0 R_0 T_0}{\gamma_0} \tag{137}
\]
Elimination of \( p_0 \) between (136) and (137) gives the relation

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

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 \((8.31434 \times 10^7 \text{ erg/mole }^0K\)) , \( P_C \) the reservoir pressure in dyne/cm\(^2\), \( SM(T_0) \) the nondimensional sonic mass flux computed by NRMAX, and \( m_* \) the sonic mass flux \( (\text{SMASS}, \frac{g}{\text{cm}^2\text{sec}}) \) based on the input total mass flow \( M \). Equation (138) may be rewritten

\[
T_0 = \frac{W_0(T_0)}{C} \left[ \frac{SM(T_0)}{C} \right] \cdot \frac{F(T_0)}{C}
\]

where the parameter \( C \) (denoted by \( \text{CONST} \) in the program) is a constant of the iteration:

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

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

\[
\frac{F_2 - F_1}{T_0 - T_2} = \frac{F_2 - F_1}{T_2 - T_1}
\]
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:

$$F = D_1 + D_2 T_0 + D_3 T_0^2$$  \hspace{1cm} (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

$$A \cdot D = B$$  \hspace{1cm} (144)

where

$$A = \begin{bmatrix}
1 & T_1 & T_1^2 \\
1 & T_2 & T_2^2 \\
1 & T_3 & T_3^2 \\
\end{bmatrix}$$ \hspace{1cm} (145a)

$$B = \begin{bmatrix}
F_1 \\
F_2 \\
F_3 \\
\end{bmatrix}, \quad D = \begin{bmatrix}
D_1 \\
D_2 \\
D_3 \\
\end{bmatrix}$$ \hspace{1cm} (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

$$T_0 = C \left( D_1 + D_2 T_0 + D_3 T_0^2 \right)$$ \hspace{1cm} (146)

The solution of this quadratic equation for $T_0$ is
\[ T_0 = \frac{1}{2D_3} \left[ - \left( D_2 - \frac{1}{C} \right) + \sqrt{\left( D_2 - \frac{1}{C} \right)^2 - 4D_1D_3} \right] \]  
(147)

The sign before the radical is determined by noting that, for \( D_3 \rightarrow 0 \), the solution must approach 2. 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,

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

\[
\begin{align*}
 u_* &= a_* = \sqrt{\frac{2}{\gamma + 1}} \quad a_0 \\
 a_0 &= \sqrt{\frac{\gamma p}{\rho}} = \frac{\gamma R_0 p_0}{w_0} = \sqrt{\gamma - 1} \quad h_0
\end{align*}
\]  

(149b)  

(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:

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

(150)

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:

\[
p_0 = 0.743 \quad m_* \quad \sqrt{h_0}
\]

(151)

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:

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

(152)

which is the form used in RESTMP.

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°C. 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 \( HS \) (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:

\[ T_0 = T_1 \cdot \frac{h_S}{h_1} \]  \hspace{1cm} (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:

\[ T_0 = T + (h_S - h) \frac{(T_p - T)}{(h_p - h)} \]  \hspace{1cm} (154)

This iteration on temperature is continued until the calculated enthalpy is equal to the desired value to within 1 part in 10^4. 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/cm^2 sec 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 $T_e/T_0$, and $CHA = h_0 W_0/R_0 T_0$. 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 G1(J), G2(J), G3(J), G4(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 

ORIGIN ALL PAGE 1B
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

\[ P(J) = 2 \cdot \frac{F3(J) - F2(J)}{F2(J) - F1(J)} \]  

(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 \((f_2-f_1)/f_1 = X4(J)/F1(J)\) 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) \( \geq 0 \), 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) \( \leq -1.25 \), these quantities are calculated from equations I(398). Since the Fortran P(J) corresponds to \((-P \Delta x)\) in equations I(398), these formulas can be rewritten in the form
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 which can easily be derived by substituting the Maclaurin expansion for \( e^x \) into (156). In RNKT, the sums (157) for \( F_1, F_2, F_3 \) 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:

\[
(T_n)_k = (T_n)_{k-1} \cdot \frac{P(J)}{n+k} \tag{158}
\]

where \( (T_n)_k \) denotes the \( k^{th} \) term in the expansion for \( F_n. \) In RNKT, 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 $T_3$ to $S_3$ no longer affects the value of $S_3$ to within floating-point accuracy.

Next, the $GJ_4(J)$ array is set. For each $J$, if $P(J)$ is negative, $GJ_4(J)$ is computed from equation I(405). For zero or positive $P(J)$, $GJ_4(J)$ 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 $GJ_4(J)$, $CX$ is set to $CXB + \text{DELTAX}$, and a final call to DERIVS is executed. The resulting values for the derivatives of the flow variable are loaded into $F_4(J)$, 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 Tremain'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 $M$, $\text{DEL}$ the flow deflection angle (i.e., the angle of attack of the wedge surface, $\alpha$), and $\text{GAM}$ 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 $\sin^2 \sigma$:

*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.
\[ \sin^6 \sigma + b \sin^4 \sigma + c \sin^2 \sigma + d = 0 \]  \hspace{1cm} (159)

where

\[ b = -\frac{M^2+2}{M^2} - \gamma \sin^2 \alpha \]  \hspace{1cm} (160a)

\[ c = \frac{2M^2+1}{M^4} + \left[ \frac{(\gamma+1)^2}{4} + \frac{\gamma-1}{M^2} \right] \sin^2 \alpha \]  \hspace{1cm} (160b)

\[ d = -\frac{\cos^2 \alpha}{M^4} \]  \hspace{1cm} (160c)

Equation (159) has three roots 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

\[ F(u) = u^3 + bu^2 + cu + d \]  \hspace{1cm} (161a)

\[ F'(u) = 3u^2 + 2bu + c \]  \hspace{1cm} (161b)

an improved estimate \( u' \) of the root is obtained, at each step of the iteration, using the formula

\[ u' = u - \frac{F(u)}{F'(u)} \]  \hspace{1cm} (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 $F(u) = 0$:

$$(u-u_1) (u^2 + Au + 3) = C \quad (163)$$

Equating the coefficients of corresponding powers of $u$ in (161a) and (163) gives

$$A = u_1 + b \quad (164a)$$

$$B = -d/u_1 \quad (164b)$$

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 $A^2 - dB$ is positive, the lower of the two values (165), $u_2$, is equal to $\sin^2 \sigma$ for the weak shock.

After the shock angle $\sigma$ has thus been determined, subroutine 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,
\[
\frac{p_2}{p_1} = \frac{2 \gamma M^2 \sin^2 \sigma - (\gamma - 1)}{\gamma + 1}
\]

(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. 1), 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 comment cards 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 D@ loop ending at statement 110. For each species J, the properties are computed at a preset series of temperatures \(T_T(I)\) up to the input value of the reservoir temperature \(T_D\). These temperatures are as follows:

- From \(100^\circ\text{K}\) to \(1000^\circ\text{K}\) at intervals of \(100^\circ\text{K}\)
- From \(1000^\circ\text{K}\) to \(3000^\circ\text{K}\) at intervals of \(200^\circ\text{K}\)
- From \(3000^\circ\text{K}\) to \(10000^\circ\text{K}\) at intervals of \(500^\circ\text{K}\)
- From \(10000^\circ\text{K}\) to \(20000^\circ\text{K}\) at intervals of \(1000^\circ\text{K}\)
- From \(20000^\circ\text{K}\) to \(50000^\circ\text{K}\) at intervals of \(2000^\circ\text{K}\)
- From \(50000^\circ\text{K}\) to \(125000^\circ\text{K}\) at intervals of \(5000^\circ\text{K}\)
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°C.)

The thermal properties are calculated by calling subroutine THERM for 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°C below the switchover temperature, CMAXX-CTAP. The thermo fit is used for temperatures more than 500°C above the switchover temperature. In a range 1000°C 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°C below CMAXX times the reservoir temperature. The results of the thermal property calculations based on the physical model are stored in the array TPRP(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 TPRP 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°C above CMAXX-CTAP, so that the calculations in THERM are based solely on the thermo fit. The results are stored in TPRP(I,K,M) for M = 2.

Finally, all of the results for the species are printed out. The properties computed and printed are
TPR0P(I,1,M) = $\frac{H_j^0 - H_{j0}^0}{R_0T}$ (chemical potential in excess of the formation enthalpy, divided by $R_0T$)

TPR0P(I,2,M) = $H_j - H_{j0}$ (molar enthalpy in excess of the formation enthalpy, kcal/mole)

TPR0P(I,3,M) = $C_{pj}$ (molar heat capacity, cal/mole$^0$K)

TPR0P(I,4,M) = $S_j^0$ (molar entropy, cal/mole$^0$K)

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 D0 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 Lth step will be applied. The indices of the species belonging to such a species pair are I1QJ, J1QJ (referred to the
master list of species).* For each step, STUNT2 prints out L, I, the index $KQ(I)$ 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 DO 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 DO 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 $L^{th}$ 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 $L^{th}$ 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 $L^{th}$ 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, STUNTP 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, STUNTP prints a table giving the transport cross sections for the various species pairs as functions of temperature. If ISW1B = -1, STUNTP 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 STUNTP, 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):

\[
Q(1,I,J) = \Omega_{ij}^{(1,1)} \\
Q(2,I,J) = \Omega_{ij}^{(2,2)} \\
Q(3,I,J) = B_{ij} \Omega_{ij}^{(1,1)}
\]
They are expressed in units of $10^{-16}$ cm$^2$.

3.64 Subroutine THERM

THERM is the routine which computes the species thermal properties: $\mu_j^\circ$ (chemical potential at standard pressure), $H_j$ (molar enthalpy), $S_j^\circ$ (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

\[
\begin{align*}
X MJ J (J) &= \frac{\mu_j^\circ}{R_0 T} \\
SHJ (J) &= \frac{H_j}{R_0 T_0} \\
SEN T (J) &= \frac{S_j^\circ}{R_0} \\
CC PJ (J) &= \frac{C_{pj}}{R_0}
\end{align*}
\]

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 $J^{th}$ 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 \( \text{CT} \), 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 \text{THERM}, the thermo fit coefficients \( \text{TFA}(J) = a \), etc., have been nondimensionalized in subroutine \text{INIT} so as to allow direct use of the nondimensional temperature \( \text{CT} \) 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 \( \text{INEQV} \) 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 THRAT

The function of THRAT 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 GEOMAR, 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 AGSOLN. 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 THRAT.

In addition to calculating RSA, THRAT 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 \( N = \text{ISS} \) to be used in the transport calculations and then calls subroutine \textsc{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, \textsc{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

\[ ZM1(I) = \frac{128 \times 10^{-24}}{9 \pi N_0 k^3} \quad W_i = 2.8567 \frac{\text{cm}^2 (\text{OK})^3}{\text{milliwatts}^2 (\text{k})^4} \quad \text{(171a)} \]

\[ ZM2(I) = \frac{4}{15} N_0 k \times 10^7 \quad W_i = 0.032064 W_i \quad \text{g} \text{OK}/\text{joule} \quad \text{(171b)} \]

\[ C(I,J) = \frac{8 \times 10^{-12}}{3k} \sqrt{\frac{2 W_i W_j}{\pi N_0 k (W_i + W_j)}} \quad \text{for} \ I \leq J \quad \text{(171c)} \]

\[ C(I,J) = \frac{W_i - W_j}{W_i + W_j} \quad \text{for} \ I > J \quad \text{(171d)} \]

Once the gas species have been defined by calling \textsc{transx}, the transport properties for any given gas condition are computed by calling \textsc{transp} (\texttt{ttab}, \texttt{p}), where \texttt{ttab} is the gas temperature and \texttt{p} is the gas pressure in atmospheres. The species mole fractions are obtained from the array \texttt{savec(j)} in common block \texttt{/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 \textsc{therm1} of subroutine \textsc{therm}. Input to the calculation is by means of the common variable \texttt{ct} which specifies the ratio between the temperature \texttt{ttab} for which the specific heats are to be calculated and the reservoir temperature \texttt{ctap}. The nondimensional molar heat
capacities at constant pressure $W_{c_{pj}}/N_0$ 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 (95), and are also used to compute the nondimensional frozen specific heat $C_P^{\theta T} = W c_{pf}/N_0 k$ for the mixture, which is required in the calculations of the Prandtl and Lewis numbers, equations (83) and (84). In calculating the specific heats, it is necessary to save the previous values of $C_T$ 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 $\Omega_{ij}(1,1), \Omega_{ij}(2,2),$ 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 (88) to (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:

\begin{align*}
Q(1, I, J) &= \frac{1}{k} \sqrt{T} \Delta_{ij}^{(1)} (cm (^0K)^{3/2} \text{milliwatt}) \quad \text{for } I \leq J \quad (172a) \\
Q(1, I, J) &= \frac{25}{4} \sqrt{T} a_{ij}^{(K)} (cm (^0K)^{3/2} \text{milliwatt}) \quad \text{for } I > J \quad (172b) \\
Q(2, I, J) &= \frac{25}{8} \sqrt{T} ZM2(I) A_{ij}^{(\mu)} (cm (^0K)^{3/2} \text{milliwatt}) \\
&= \frac{5}{6} \frac{1}{k} \sqrt{T} \Delta_{ij}^{(2)} \quad \text{for } I \leq J \quad (172c)
\end{align*}
\[ Q(2, I, J) = \frac{25}{8} \sqrt{\frac{\tau}{T}} a_{ij}^{(\mu)} \left( \frac{\text{cm} \ (\text{oK})^{1/2}}{\text{millipoise}} \right) \text{ for } I > J \quad (172d) \]

\[ Q(3, I, J) = \frac{25}{8} \sqrt{\frac{\tau}{T}} A_{ij}^{(K)} \left( \frac{\text{cm} \ (\text{oK})^{3/2}}{\text{milliwatt}} \right) \quad (172e) \]

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 \( \rho c_p f D_{ij}/\sqrt{T} \) in milliwatts/cm \( (\text{oK})^{3/2} \) is also computed and stored by the code as 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 ELC\&ND, KINT, and KANDMU. The quantities returned to TRANSF by these subroutines are

\[
\begin{align*}
\text{SIGMA} &= \sigma \text{ in mhos/cm} \\
\text{ZKINT} &= K_{\text{int}}/\sqrt{T} \quad \text{(milliwatts/cm-(oK)}^{3/2} \right) \\
\text{ZK(1)} &= K_{\text{tr}}/\sqrt{T} \quad \text{(milliwatts/cm-(oK)}^{3/2} \right) \quad (173) \\
\text{ZK(2)} &= \mu/\sqrt{T} \quad \text{(millipoise/(oK)}^{1/2} \right)
\end{align*}
\]

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 TRANSF according to equations I(83), I(84) and I(94) to obtain the viscosity 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 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 /OUTPUT/. 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 equivalence of the variable names used in WEDGE to the corresponding 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.
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 WESOLIN in double precision, and by using an analytical series solution for $\zeta \leq 10^{-8}$:

$$z = 1.65096 \, \zeta^{2/3} + 0.50869 \, \zeta^{5/6} - 0.0249 \zeta$$  \hspace{1cm} (174a)

$$zz' = 1.81711 \, \zeta^{1/3} + 1.25975 \, \zeta^{1/2} + 0.14712 \, \zeta^{2/3}$$  \hspace{1cm} (174b)

$$(zz')' = 0.6057 \, \zeta^{-2/3} + 0.62988 \, \zeta^{-1/2} + 0.09808 \, \zeta^{-1/3}$$  \hspace{1cm} (174c)

Equation (174a) was derived by Boger 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

$$\lambda = zz' = (6 \, \zeta)^{1/3} + (3 \, \zeta^{1/2}$$  \hspace{1cm} (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

$$\lambda' = \lambda - (\zeta - \zeta_1) \frac{d \lambda}{d \zeta}$$  \hspace{1cm} (176)

in which $\zeta_1$ is the input value ZETA and, from I(480) with $\Gamma = 0$ and I(483),
\[ \frac{d \lambda}{d \zeta} = \frac{d}{d \zeta} \left( z \frac{dz}{d \zeta} \right) = \frac{1 + \sqrt{zz'}}{z} = \frac{1 + \sqrt{\lambda}}{z} \]  \hspace{1cm} (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: \( \text{zz}' \) (denoted by ZZP) is obtained from equation I(491), \( (\pi')' \) (denoted by ZZPP) from I(434) and I(485), and \( z \) (denoted by Z) from I(492).

3.6. \textbf{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?
  yes (140) Can this cross section be treated as ≤ N?
  no (300) Delete species > N from edited cross section data

(Below 380) Add Default data for missing cross sections. Can all required data be supplied?

(2nd below 150) Increase N by 1; revise I(J) to include required species in cross section computations

(190) Revise data to make the cross section ≤ N

(540) Write diagnostic; terminate case.

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 TRASP 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, NO(L) in the edited data is taken to be the sum from 1 to L of the original NNO(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 /TRANS2/, 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 for the pair $(LL,J)$ 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 $Q(1,II,JJ)$ 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(I,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 300 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 ≠ 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 KQ = 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 ≤ 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 ≤ 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
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 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 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 LL′th 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 state-
ment 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 spec-
ies 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 NKQ^th 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 appropi-
ate for the effective Coulomb potential I(100) from
the preset array VC011, 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 origi-
ally in the loop ending at statement 340, is also
revised to conform to the raw 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 ILELEC = 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, independ-
ent of the values of the indices.
**TABLE III**

**INDICES USED FOR CALCULATION OF EFFECTIVE COULOMB CROSS SECTIONS IN SUBROUTINE XSECT**

<table>
<thead>
<tr>
<th>Passage number through loop</th>
<th>Species in cross section data are supplied*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Species pairs</td>
<td></td>
</tr>
<tr>
<td>(1,1)</td>
<td></td>
</tr>
<tr>
<td>(e,e) (e,1)</td>
<td></td>
</tr>
<tr>
<td>(2,2)</td>
<td></td>
</tr>
<tr>
<td>(1,2)</td>
<td></td>
</tr>
<tr>
<td>(e,2)</td>
<td></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 $\vec{v}_{i-c,j}$ for expressing the dependent species in terms of the $c$ independent species

CAPX(J) Mole fraction $X_J$ for the $J^{th}$ species

GJ(J) Species concentrations in moles per gram of mixture; equivalenced to CAPX(J)

$A$ $\alpha$, constant in the density-area relation I(383)

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
AR  Indicator (initially 0) which is set to 1 when $\Delta \ln A_e/\Delta x$ becomes positive downstream of the throat in the nonequilibrium solution by the inverse method.

ARBA  Maximum allowable number of tries at switching from upstream to downstream region in nonequilibrium solution

ARBB  Counter for number of times upstream-downstream switching point is moved downstream

BZERØ  $b_0/w^0$, 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.

C  Constant C in density-area curvefit relation I(383)

CARB  Effective area ratio at the beginning of the current integration step

CH  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

CHA  Nondimensional specific enthalpy in the reservoir

CLNT  $\ln T$, where $T$ = temperature in $^0K$

CM  Molecular weight of gas mixture, $W$ (g/mole)

CMA  Molecular weight in reservoir, $W_0$ (g/mole)

CRA  $R_0$, universal gas constant (1.9872 cal/mole $^0K$)

CRP  $R_0T_0$, where $T_0$ = reservoir temperature ($^0K$)

CRRB  Intermediate variable in calculation of entropy
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRS</td>
<td>Dummy variable for entropy</td>
</tr>
<tr>
<td>CSTA</td>
<td>$0.5 \ln \left( \frac{W_0}{R_0 T_0} \right)$, where $W_0$ is the reservoir molecular weight (g/mole), $R_0$ the universal gas constant, and $T_0$ the reservoir temperature (°K)</td>
</tr>
<tr>
<td>CT</td>
<td>Ratio of the temperature to the reservoir temperature, $T/T_0$</td>
</tr>
<tr>
<td>CTAP</td>
<td>Reservoir temperature, $T_0$ (°K)</td>
</tr>
<tr>
<td>CTB</td>
<td>Dummy variable for CT</td>
</tr>
<tr>
<td>CTC</td>
<td>Dummy variable for CT</td>
</tr>
<tr>
<td>CTMAX</td>
<td>Ratio of throat temperature to reservoir temperature</td>
</tr>
<tr>
<td>CTMXK</td>
<td>Temperature for switching from thermo-fit to physical model for species properties, divided by reservoir temperature</td>
</tr>
<tr>
<td>CTP</td>
<td>Temperature in °K</td>
</tr>
<tr>
<td>CTPL</td>
<td>$\ln T_0$</td>
</tr>
<tr>
<td>CTT</td>
<td>CT at the last printed step of the nonequilibrium solution</td>
</tr>
<tr>
<td>CX</td>
<td>Streamwise coordinate $x$ (cm), zero at the throat and positive downstream</td>
</tr>
<tr>
<td>CXB</td>
<td>CX at the beginning of the current integration step</td>
</tr>
<tr>
<td>CXMAX</td>
<td>Maximum allowable value of $x$ (cm)</td>
</tr>
<tr>
<td>DATEST</td>
<td>Value of effective area ratio at which nonequilibrium solution is switched from upstream to downstream region</td>
</tr>
<tr>
<td>DBTEST</td>
<td>(Not used)</td>
</tr>
</tbody>
</table>
DELT1  Nondimensional temperature decrement used in frozen and equilibrium solutions
DELT2  (Not used)
DELTAX  Increment in x (cm)
DLLOGA  \( \frac{d \ln a_e}{dx} \)
DLLOGR  \( \frac{d \ln (\rho/\rho_0)}{dx} \), logarithmic derivative of nondimensional density
DT     \( \frac{d(T/T_0)}{dx} \), gradient of nondimensional temperature
ENT     Intermediate variable used in frozen flow solution
FLUX    Nondimensional mass flux, \( (\rho/\rho_0) \cdot u/\sqrt{R_0 T_0/\dot{W}_0} \)
HDELX   \( \text{DELTAX}/2 \)
PCT     \( \delta(T/T_0) \), perturbation in nondimensional temperature
PCTEST  Tolerance on \( \chi \) when testing size of \( \delta \chi_i \)
PRES    Nondimensional pressure, \( p/p_0 \)
PRESA   Reservoir pressure \( p_0 \) (atm)
PRESB   Nondimensional pressure at the preceding flow point; used in Mach number calculation
PRESTH  Nondimensional pressure at throat
PRHO   Perturbation in nondimensional density, \( \delta \rho/\rho_0 \)
RHAP   Density in reservoir, \( \rho_0 \) (g/cm³)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RH$\bar{\rho}$</td>
<td>Nondimensional density, $\rho/\rho_0$</td>
</tr>
<tr>
<td>RH$\bar{\rho}B$</td>
<td>RH$\bar{\rho}$ at the preceding flow point; used in Mach number calculation</td>
</tr>
<tr>
<td>RH$\bar{\rho}$BAR</td>
<td>Effective density in high-density equation of state divided by effective reservoir density</td>
</tr>
<tr>
<td>RH$\bar{\rho}$C</td>
<td>(Not used)</td>
</tr>
<tr>
<td>RH$\bar{\rho}$P</td>
<td>Density $\rho$ in gm/cm$^3$</td>
</tr>
<tr>
<td>RH$\bar{\rho}$L</td>
<td>$ln \rho$</td>
</tr>
<tr>
<td>RHTH</td>
<td>Nondimensional density at throat, $\rho_*/\rho_0$</td>
</tr>
<tr>
<td>R$\bar{\rho}$BARA</td>
<td>$\bar{\rho}_0/\rho_0$</td>
</tr>
<tr>
<td>R$\bar{\rho}$BARP</td>
<td>Effective density in reservoir, $\bar{\rho}_0$ (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 $\text{^oK}$</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, $\lambda$, 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 \sum_j \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\emptyset 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 \text{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 \chi^*_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)}
CCPJ(J) Molar heat capacity of the Jth species divided by R₀
CEACT(I) Activation energy for ith reaction (cal/mole ÒK)
CGI(I) Species molecular weights (g/mole)
CGMU(I) ln Xᵢ, logarithms of mole fractions
CHI(I) χᵢ, variable measuring the departure of the ith reaction from equilibrium, eq. I(289)
CHII(I) Intermediate variables in computation of equilibrium constant based on mole fractions
CLN1MC(I) ln (1 - χᵢ)
CLNPI(I) ln Pᵢ, variables in nonequilibrium solution
CMW(I) Atomic weights of elements (g/mole)
DGJ(J) d Yᵢ/dx; equivalenced to GJA(J)
ETAI(I) ηᵢ, temperature exponent in rate constant for ith reaction
ETAJ(J) nᵢ, number of atoms in a molecule of the Jth species; 0 value indicates no physical model data for thermal properties
GJA(J) γᵢ₀, concentration of jth species in reservoir
GJB(J) GJ(J) at the beginning of the current integration step
PERTGJ(J) δγᵢ, perturbations in species concentrations
PGJ(J) Intermediate variables in Newton-Raphson calculation of equilibrium composition.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PI(I)</td>
<td>Variables $P_i$ in nonequilibrium solution, eq. I(288)</td>
</tr>
<tr>
<td>PICH(I)</td>
<td>$P_i \chi_i$</td>
</tr>
<tr>
<td>QM(I)</td>
<td>Number of moles $q_j$ of the $j^{th}$ independent species in one mole of mixture</td>
</tr>
<tr>
<td>QQ(I)</td>
<td>1 if a third-body list is provided for the $i^{th}$ reaction, 0 if not</td>
</tr>
<tr>
<td>SAJ(J)</td>
<td>$b_j + (1.5 + n_j) \ln T_0$, constant in physical-model expression for $\mu_j^{o}$</td>
</tr>
<tr>
<td>SBJ(J)</td>
<td>$b_j$, eqs. (51), Vol. I; equivalenced to SAJ(J)</td>
</tr>
<tr>
<td>SDCHI(I)</td>
<td>$\delta \chi_i$, perturbation in $\chi_i$</td>
</tr>
<tr>
<td>SDGJ(J)</td>
<td>Change in $\gamma_j$ over an integration step</td>
</tr>
<tr>
<td>SENT(J)</td>
<td>Nondimensional species entropy, $S_j^{o}/R_0$</td>
</tr>
<tr>
<td>SHJ(J)</td>
<td>Nondimensional species enthalpy, $H_j^{o}/R_0 T_0$</td>
</tr>
<tr>
<td>SHJA(J)</td>
<td>Nondimensional formation enthalpy of $j^{th}$ species, $H_{j0}^{o}/R_c T_0$</td>
</tr>
<tr>
<td>SHJAP(J)</td>
<td>Formation enthalpy of $J^{th}$ species, cal/mole</td>
</tr>
<tr>
<td>SKIL(I)</td>
<td>Intermediate variable in calculation of dependent species concentrations from concentrations of independent species</td>
</tr>
<tr>
<td>SS(J)</td>
<td>$\ln \gamma_{j0}$, logarithms of reservoir species concentrations</td>
</tr>
<tr>
<td>TB(J)</td>
<td>Temporary storage for quantities necessary to restart a step in the nonequilibrium calculation</td>
</tr>
<tr>
<td>TFA(J)</td>
<td>$a_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^{o}$</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TFB(J)</td>
<td>$b_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$</td>
</tr>
<tr>
<td>TFC(J)</td>
<td>$c_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$</td>
</tr>
<tr>
<td>TFD(J)</td>
<td>$d_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$</td>
</tr>
<tr>
<td>TFE(J)</td>
<td>$e_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0/T$</td>
</tr>
<tr>
<td>TFK(J)</td>
<td>$k_j$, coefficient in thermo-fit expressions for $H_j$ and $\mu_j^0$</td>
</tr>
<tr>
<td>THEV(J)</td>
<td>Nondimensional characteristic vibrational temperature, $\theta_{v_j}/T_0$</td>
</tr>
<tr>
<td>THEVP(J)</td>
<td>Characteristic vibrational temperature for $J^{th}$ species, $\theta_{v_j} (^\circ K)$</td>
</tr>
<tr>
<td>XMJAT(J)</td>
<td>$\mu_j^0/R_0T$</td>
</tr>
<tr>
<td>XNUI(I)</td>
<td>$\nu_i = \sum_j \nu_{ij}$, sum of stoichiometric coefficients on the reactant side of the $i^{th}$ reaction</td>
</tr>
<tr>
<td>BETA(I,J)</td>
<td>$\beta_{ij} = \nu'<em>{ij} - \nu</em>{ij}$</td>
</tr>
<tr>
<td>BTA(I,J)</td>
<td>Equivalent to BETA(I,J); not used</td>
</tr>
<tr>
<td>ELJ(L,J)</td>
<td>Nondimensional energy of $L^{th}$ electronic level of $J^{th}$ species, $E_{Lj}/R_0T_0$</td>
</tr>
<tr>
<td>GELJ(L,J)</td>
<td>Degeneracy of $L^{th}$ electronic level of $J^{th}$ species, $g_{Lj}$</td>
</tr>
<tr>
<td>XNUIJ(I,J)</td>
<td>$\nu_{ij'}$, stoichiometric coefficient of $J^{th}$ species on reactant side of $i^{th}$ reaction</td>
</tr>
<tr>
<td>XNUIJP(I,J)</td>
<td>$\nu_{ij''}$, stoichiometric coefficient of $j^{th}$ species on product side of $i^{th}$ reaction</td>
</tr>
<tr>
<td>IC</td>
<td>Number of ions included in chemical model</td>
</tr>
</tbody>
</table>
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 (v) 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)
ISM CNR  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
<table>
<thead>
<tr>
<th>ISW1A</th>
<th>Input control variable; 0 suppresses frozen solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISW1B</td>
<td>Input control variable; nonzero value gives edits of transport cross section calculations</td>
</tr>
<tr>
<td>ISW2A</td>
<td>Input control variable; 0 suppresses non-equilibrium solution</td>
</tr>
<tr>
<td>ISW2B</td>
<td>Input control variable; selects types of input data for reservoir calculation</td>
</tr>
<tr>
<td>ISW3A</td>
<td>Input control variable; 0 suppresses equilibrium solution</td>
</tr>
<tr>
<td>ISW3B</td>
<td>Input control variable; 0 suppresses boundary layer calculations</td>
</tr>
<tr>
<td>ISW4A</td>
<td>Input control variable; must be nonzero if another case follows, 0 for last case</td>
</tr>
<tr>
<td>ISW4B</td>
<td>Input control variable; nonzero gives dump in boundary layer routine BLAYER</td>
</tr>
<tr>
<td>ISW5A</td>
<td>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</td>
</tr>
<tr>
<td>ISW5B</td>
<td>Input control variable; nonzero gives dump in subroutines EXACT, COMM, NONEQ and PRTA for debugging nonequilibrium calculations</td>
</tr>
<tr>
<td>ISW6A</td>
<td>Input control variable; if positive, only the reservoir equilibrium calculation is done; if negative, edit of species thermal properties is produced</td>
</tr>
<tr>
<td>ISW6B</td>
<td>Input control variable; 0 suppresses the output of species mole fractions in the free-stream and model-point output</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>IUPD</td>
<td>Indicator (initially 1), reset to 0 when switch from inverse method to direct integration is made in nonequilibrium solution</td>
</tr>
<tr>
<td>IZERØ</td>
<td>0 (set in READ)</td>
</tr>
<tr>
<td>JJK</td>
<td>Indicator (initially 0), reset to 1 in NEWRAP when electrons are dropped from the equilibrium calculation</td>
</tr>
<tr>
<td>LC</td>
<td>(Not used)</td>
</tr>
<tr>
<td>M1</td>
<td>Index equal to c+l</td>
</tr>
<tr>
<td>NFIT</td>
<td>Indicator, 0 if thermo fit data are not used for any species</td>
</tr>
<tr>
<td>NIT</td>
<td>Counter for number of iterations in Newton-Raphson procedure</td>
</tr>
<tr>
<td>NNN</td>
<td>Integration step counter</td>
</tr>
<tr>
<td>NNS</td>
<td>Number of times Δx has been increased by current value of SC</td>
</tr>
<tr>
<td>NQS</td>
<td>Number of successful integration steps before Δx is increased</td>
</tr>
<tr>
<td>NQT</td>
<td>(Not used)</td>
</tr>
<tr>
<td>NTEST</td>
<td>Number of iterations allowed in Newton-Raphson procedure</td>
</tr>
<tr>
<td>IGJ(J)</td>
<td>Indicator; 1 if thermo-fit data are used for Jth species, 0 if not</td>
</tr>
<tr>
<td>IGM(J)</td>
<td>Number of electronic levels for the Jth species</td>
</tr>
<tr>
<td>ITB(I)</td>
<td>Temporary storage for indices necessary to restart step in nonequilibrium calculation</td>
</tr>
</tbody>
</table>
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 I\(^{th}\) species

ACQ\( \Omega \)(I)  Hollerith description for case

EIMENT(I)  Chemical symbols for elements (Hollerith)

HP(I)  Symbols for species (Hollerith)

4.2 Common /AEG\( \Omega \)M/

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

NSECT(I)  Number of sections in the I\(^{th}\) profile

NSECTU(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)

XZERO  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)

VISR\(\omega\)T  Ratio of viscosity in the reservoir to the reservoir temperature (poise/\(^{\circ}\)K)

DIAM(L)  Throat diameter for the \(L^{th}\) profile (inches)

\(S_W = (h_w/h_0)^{-1}\), where \(h_w\) is the gas enthalpy at the wall and \(h_0\) the free-stream stagnation enthalpy
Parameter | Description
--- | ---
RO | Characteristic length used in boundary layer calculations (cm); equal to the throat radius for two-dimensional and axisymmetric nozzles; equal to $\sqrt{\frac{A^*_I}{\pi}}$ for channels, where $A^*_I$ is the geometric cross sectional area of the channel at the throat.
JDIM | Dimensionality index, 0 for two-dimensional nozzles, 1 for axisymmetric nozzles.
IP\$INT | Index of flow solution points at which boundary layer calculations are done.

4.6 **Common/BLNE/**

X(I) | For $i = 1$ and 2, the streamwise coordinate along the nozzle or channel surface for the $I^{th}$ profile (cm).
RDINP(I) | The quantity $\vartheta$ for the boundary layer on the $I^{th}$ profile at the preceding flow point, eq. I(171).
DELBLP(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.
C\$UPLD | 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 \(L^{th}\) profile, divided by \(R_0\)

**SN(L)**  Correlation parameter \(n\), eqs. I(158) and I(174), for the boundary layer on the \(L^{th}\) profile

**XSN(L)**  Averaged correlation parameter \(\bar{n}\), eq. I(217), for the boundary layer on the \(L^{th}\) profile

**PRREF**  Prandtl number at the reference temperature

**HR**  Recovery enthalpy \((\text{cal/g})\)

**QWDOT(L)**  Heat flux to the wall surface with the \(L^{th}\) profile \((\text{Btu/ft}^2\text{sec})\)

**TAUW(L)**  Shear stress on the wall surface with the \(L^{th}\) profile \((\text{lb-f/ft}^2)\)

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 Index of second channel profile in compiled-in list of profiles

I = 3 Hollerith name for channel

I = 4 Index (1 or 2) of the profile which diverges from the channel axis least rapidly downstream of the throat

I = 5 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
Conversion factor from nondimensional velocity SU to velocity in ft/sec

Conversion factors equal to 1

Conversion factor from viscosity in poise to viscosity in lbm/ft sec

Conversion factor equal to 1

Conversion factor from length in cm to length in inches

Conversion factor from nondimensional enthalpy CH to enthalpy in Btu/lbm

Conversion factor equal to 1

Conversion factors from nondimensional boundary layer thicknesses DELBL and THETA to thicknesses in inches

Conversion factors equal to 1

\[(\sqrt{\rho_0 T_0})^{-1} \sum H_j \frac{\gamma_j}{dx}, \text{ where } H_j \text{ is the molar enthalpy of the } j\text{th species and } \gamma_j \text{ is the concentration of the } j\text{th species in moles per gram of mixture (set in subroutine EXACT)}\]

Data for first input-defined element
- J = 1 Atomic number
- J = 2 Atomic weight (g/mole)

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 EQC 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, $(R_0T_0)^{-1} \sum x_j H_j$
ZCM Mean molecular weight
ZRBP Effective density $(\bar{\rho})$ in reservoir for imperfect-gas model
ZRHO 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
4.18 **Common /ERROR/**

ERR  Logical flag set to "true" in DUMP routine; if "true", control is passed back to Main program, DUMP.EX is called to print a dump of common data, and case is terminated

4.19 **Common /GLIM2/**

NÆREAC(I) Logical control variables; when NÆREAC(I) is set to .TRUE., the Ith reaction is suppressed

4.20 **Common /INGNE/**

A′(K,I) The inverse A′ of the square submatrix A′ of the matrix A, specifying the number of atoms of the jth element per molecule of the ith species; see eq. I(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
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 $\alpha$ 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 ($^\circ$K)</td>
</tr>
<tr>
<td>XMODPL</td>
<td>Initial distance from the throat at which model condition calculations are to be done (inches)</td>
</tr>
<tr>
<td>DXMLDP</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 ($^\circ$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>MODLPPT</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 - δ*/R0 for the Lth profile at the sonic point

DDELBL(L) (d δ*/dx)/R0, 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
ISHAPE = 3  Circular section concave toward axis

I = 17-27  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 $P_1$, $P_2$, $P_3$ for each profile. For ISHAPE = 1, the equation of a straight profile is

$$r(x) = P_1 + P_2x$$

For ISHAPE = 2 or 3

$P_1 =$ distance of circle center from axis

$P_2 =$ x-coordinate of circle center

$P_3 =$ circle radius

I = 64  Hollerith facility name

ZP2(I)  Specifications of standard profile no. 2

ZP20(I)  Specifications of standard profile no. 20

4.30  Common [QUIMID]

TB7  $N_0 \rho_0$, where $N_0$ is Avogadro's number (6.0225 x 10^23 particles per mole) and $\rho_0$ 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 OUTL (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/ft²·sec)
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/ft²·sec)
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

GJM£(J) (Not used)
4.32 **Common /PŁYAT/**

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)

NÖZŻØ Nozzle profile index for preceding case

ICHAÑØ 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

NÔmRAN 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/

FLO\(\dot{\text{n}}\)

Total mass flow in lbm/sec

FACNAM

Hollerith facility name

CHANAM

Hollerith channel name

LIM\(\dot{\text{O}}\)UT

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/

ISW8B

Index controlling diagnostic dumps in the transport property routines. If ISW8B = 0, these dumps are omitted. If ISW8B > 0, the PUTQIN dump is produced once every ISW8B times the subroutine PUTQIN is called. If ISW8B < 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 $\zeta$
+ 3  Print both $Y_s$ and $\zeta$
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 (cm$^3$/mole sec or cm$^6$/mole$^2$ 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 nonequilibrium 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)

XZERØI: 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 II)

BZERØI: Input of constant in imperfect 1ps 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

AXIMØD: 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 $0.01 \left| \Delta \chi_i \right|_{\min} / DCHLL$.

DCHRAT  Parameter controlling the artificial increase in reaction rates in the perturbation solution to avoid premature startup of 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

SPI(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)
Square of the sound speed divided by the non-dimensional temperature $T/T_0$ (cm$^2$/sec$^2$)

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$^{\circ}K$

Initial estimate of temperature behind equilibrium normal shock ($^\circ$K)

Initial estimate of density ratio across equilibrium normal shock

Initial estimate of temperature behind frozen normal shock ($^\circ$K)

Initial estimate of density ratio across frozen normal shock

Number of successful integration steps since the last printout of conditions at a flow point

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)

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\( \text{\textasciitilde UT}\)/**

**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\( \text{\textasciitilde UT}\)** File number of binary output tape.

**NRC\( \text{\textasciitilde UT}\)** Number of records written on binary tape during current case of job.

**IFL\( \text{\textasciitilde W}\)** (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\( \text{\textasciitilde RY}\)/**

**SAVEC(J)** Species mole fractions for which transport property calculations are done.

4.50 **Common /THRT/**

**RSA** Factor by which cross sectional area of flow has been rescaled in nonequilibrium solution.
4.51 Common /TNCE/

SUMGH \[ \sum \gamma_j, \text{sum of the molar concentrations for all species except the electron (mole/g)} \]

SCPGH \[ \sum C_p \gamma_j/R_0, \text{sum of the products of the molar heat capacities and molar concentrations for all species except the electron, divided by the universal gas constant} \]

QDPR Power radiated from the gas, cal/cm\(^3\)-sec

QDPE Net power transferred to the electron gas (cal/cm\(^3\)-sec)

4.52 Common /TNDC/

XMJATD(J,L) Chemical potential at standard pressure (\(\mu_i^0\)); for the \(j\)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

CLNTD(L) Natural logarithm of the \(L\)th temperature (\(L = 1,\) heavy particle temperature; \(L = 2,\) electron temperature)

CTD(2) Nondimensional temperatures (\(L = 1, T/T_0;\) \(L = 2, T_e/T_0\))

4.53 Common /TNE/

TNL(I) 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
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).

FOM(I)  For I = 1-30, cross section values for elastic collision cross section table for current gas model.
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** $(W_0/R_0T_0)(dh_0/dx)$, rate of change of non-dimensional total enthalpy

**CHB** $W_0h_0/R_0T_0$, nondimensional total enthalpy at start of integration step

**SDCHA** $\Delta (W_0h_0/R_0T_0)$, 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 NONEQ</td>
</tr>
<tr>
<td>2</td>
<td>Test on $T_e$ in NONEQ</td>
</tr>
<tr>
<td>3</td>
<td>Test on $h_0$ in NONEQ</td>
</tr>
<tr>
<td>4</td>
<td>Test on DLGCA in NOEQ</td>
</tr>
<tr>
<td>5</td>
<td>Test on QDPE in NOEQ</td>
</tr>
<tr>
<td>11-30</td>
<td>Test on GJ(IFAIL-10) from RNKT in NOEQ</td>
</tr>
<tr>
<td>31-50</td>
<td>Test on GJ(IFAIL-30) from element conservation in NOEQ</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 NONEQ)</td>
</tr>
<tr>
<td>-1 to -ISS</td>
<td>Test on sign of GJ(-IFAIL) in RNKT (GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+1)</td>
<td>Test on sign of T in RNKT (GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+2)</td>
<td>Test on sign of T_e in RNKT (GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+3)</td>
<td>Test on sign of h_0 in RNKT (GJ2 or GJ3)</td>
</tr>
<tr>
<td>-(ISS+4)</td>
<td>Test on QDPE in RNKT (GJ2 or GJ3)</td>
</tr>
<tr>
<td>-31 to (-30+ISS)</td>
<td>Test on sign of GJ(-IFAIL-30) in RNKT (GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(31+ISS)</td>
<td>Test on sign of T in RNKT (GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(32+ISS)</td>
<td>Test on sign of T_e in RNKT (GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(33+ISS)</td>
<td>Test on sign of h_0 in RNKT (GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-(34+ISS)</td>
<td>Test on QDPE in RNKT (GJ4, Runge-Kutta)</td>
</tr>
<tr>
<td>-61 to -(60+ISS)</td>
<td>Test on sign of GJ(-IFAIL-60) in RNKT (GJ4, Treanor)</td>
</tr>
<tr>
<td>-(61+ISS)</td>
<td>Test on sign of T in RNKT (GJ4, Treanor)</td>
</tr>
<tr>
<td>-(62+ISS)</td>
<td>Test on sign of T_e in RNKT (GJ4, Treanor)</td>
</tr>
<tr>
<td>IFAIL</td>
<td>Meaning</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-(63+ISS)</td>
<td>Test on sign of $h_0$ in RNKT (GJ4, Treanor)</td>
</tr>
<tr>
<td>-(64+ISS)</td>
<td>Test on QDPE in RNKT (GJ4, Treanor)</td>
</tr>
</tbody>
</table>

4.56 **Common /TN\_NEQ/**

<table>
<thead>
<tr>
<th>CTE</th>
<th>Electron temperature divided by gas temperature in reservoir</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTE</td>
<td>Derivative of CTE with respect to $x$, i.e., $d(T_e/T_0)/dx$</td>
</tr>
<tr>
<td>BPAR</td>
<td>Parameter $b$ for reactions with $KTF = 4$</td>
</tr>
<tr>
<td>EPAR(I,JR)</td>
<td>$EPAR(1,JR) = \text{parameter } \epsilon_0 \text{ for the } JR\text{th}$ reaction in cal per $N_0$ reactions; $EPAR(2,JR) = \text{parameter } a \text{ for the } JR\text{th}$ reaction if $ITR(JR) = 1$</td>
</tr>
<tr>
<td>NT</td>
<td>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</td>
</tr>
<tr>
<td>ITR(JR)</td>
<td>Indicator of the rule for partitioning the reaction energy between the electrons, the heavy particles, and radiative losses in the $JR\text{th}$ reaction of the current gas model. See discussion in Section 4.4 of Vol. II</td>
</tr>
<tr>
<td>KTF(JR)</td>
<td>Indicator for temperature dependence of the forward rate constant $k_f$ for the $JR\text{th}$ reaction in the current gas model; see Sec. 4.4 of Vol. II</td>
</tr>
</tbody>
</table>

-253-
**KTR(JR)**  
Indicator for temperature dependence of the backward rate constant $k_r$ for the $JR$th reaction in the current gas model:

- $KTR = 0$, $k_r = 0$
- $KTR = 1$, $k_r = k_r(T)$
- $KTR = 2$, $k_r = k_r(T_e)$

**ICH(J)**  
Indicator for charged species; equal to 1 if the $J$th species is neutral, equal to 2 if the species is an ion.

**IPA(I)**  
Species index for the product atom in the $I$th reaction; if $KTF(I) \neq 4$, $IPA(I)$ is set to zero.

**IIi(K)**  
For $K = 1$ to a maximum of 5, the index of the first species in the $K$th 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.

**JJi(K)**  
For $K = 1$ to a maximum of 5, the index of the second species in the $K$th 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 $IIi(K) \leq JJi(K)$ are used.

**T**  
Temperature in °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:

\[
\begin{align*}
Q(1, I, J) & = \bar{\Omega}_{ij}^{(1,1)}, \\
Q(2, I, J) & = \bar{\Omega}_{ij}^{(2,2)}, \\
Q(3, I, J) & = B_{ij} \bar{\Omega}_{ij}^{(1,1)}.
\end{align*}
\]

At other stages in the calculations, \( Q(K, I, J) \) is also used in subroutine TRANS 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
ID2  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/

VVi(K)  For K = 1 to a maximum of 5, the input parameters for the ith step of the cross section calculations. The common block provides storage for i = 1 to 100
4.67 Common /TRPRSP/

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 /WMID/

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 \( h\omega_0/R_0T_0 \) at the throat

CMTH  Molecular weight of gas mixture at the throat

D  \( \left( \rho_\ast/\rho_0 \right)^A (A+2)^{-2} \), where \( \rho_\ast \) 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  \( \left( \rho_\ast/\rho_0 \right)^A \left[ (A+2) \ln \left( \rho_\ast/\rho_0 \right) + 1 \right] \)

ELTMIN  Case execution time in minutes
ET  Elapsed time from beginning of execution (seconds)

ET0  Elapsed time from beginning of execution at the start of a case (seconds)

GJ(I)  Species concentrations in moles/g

GJA(I)  Species concentrations in the upstream reservoir, moles/g

GMF(J)  Mole fraction of the Jth species under equilibrium sonic conditions

HOP  Reservoir enthalpy (cal/g)

I  DØ index; also, index, in the list of species for the current problem, of the Kth cold species

IRCO  Indicator; value 1 indicates that call to TRANSX was skipped in the previous case because tables of species thermal properties were generated

ISØLN  Indicator for type of flow solution:

ISØLN = 1 Frozen
ISØLN = 2 Equilibrium
ISØLN = 3 Nonequilibrium

IS3  Storage for input value of ISW3B; used to restore value after a case in which ISW3B was set to zero because NOTRAN = .TRUE.

K  Cold species index

N  Iteration counter in Newton-Raphson solution for parameter α in area-density relation

NCASE  Count for cases in a run

NCØMPL  Number of cases successfully completed in a run
Number of failed cases in a run

Nozzle profile index

Total number of records written on tape 8 during a run

1.0

3.14159

\( \ln \left( \frac{\rho^*}{\rho_0} \right) \)

A+2

\( \left( \frac{\rho^*}{\rho_0} \right)^A \)

Nondimensional flow velocity (SU) at throat

Viscosity in the reservoir (lbm/ft-sec)

4.70 Subroutine AESQLN

Displacement thickness in cm, on Lth profile

Displacement thickness at the throat in cm, on Lth profile

One-fourth of the effective flow area at the throat for a channel (cm²)

Slope dy/dx for the first profile in a channel

Slope dz/dx for the second profile in a channel

Effective ordinate of first profile in a channel (cm)

Effective ordinate of second profile in a channel (cm)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQRTAE</td>
<td>Square root of the effective area ratio, $\sqrt{A_e}$</td>
</tr>
<tr>
<td>X</td>
<td>Axial coordinate in nozzle (cm)</td>
</tr>
<tr>
<td>Y</td>
<td>Geometric ordinate of first profile in a channel (cm)</td>
</tr>
<tr>
<td>Z</td>
<td>Geometric ordinate of second profile in a channel (cm)</td>
</tr>
</tbody>
</table>

4.71 **Subroutine AGSLN**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Trial value of geometric area ratio</td>
</tr>
<tr>
<td>AE</td>
<td>Effective area ratio</td>
</tr>
<tr>
<td>AEC</td>
<td>Constant term in equation relating the geometric and effective area ratios for a channel</td>
</tr>
<tr>
<td>AG</td>
<td>Final solution for the geometric area ratio</td>
</tr>
<tr>
<td>DA</td>
<td>Dummy argument in call to GEOMAR; represents X-derivative of geometric area ratio (not used)</td>
</tr>
<tr>
<td>DEL(L)</td>
<td>Displacement thickness divided by RO, for boundary layer on Lth profile</td>
</tr>
<tr>
<td>Delp(L)</td>
<td>Displacement thickness (cm) for boundary layer on Lth profile</td>
</tr>
<tr>
<td>DSTP(L)</td>
<td>Displacement thickness at the throat (cm) for boundary layer on Lth profile</td>
</tr>
<tr>
<td>F</td>
<td>Function which is 0 when the correct geometric area ratio has been found</td>
</tr>
<tr>
<td>FØ</td>
<td>F value in previous step of iteration</td>
</tr>
<tr>
<td>ICØUNT</td>
<td>Iteration counter</td>
</tr>
<tr>
<td>L</td>
<td>Ø index over channel profiles</td>
</tr>
<tr>
<td>NAME</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>RNAME</td>
<td>Routine name for call to DUMP routine</td>
</tr>
<tr>
<td>SQRTA</td>
<td>Square root of geometric area ratio</td>
</tr>
<tr>
<td>UPD\u00f8WN</td>
<td>Indicator; 1.0 if downstream solution is desired by calling routine, -1.0 if upstream solution is desired</td>
</tr>
</tbody>
</table>

**X**
Axial coordinate in nozzle (cm)

**XL**
Temporary storage for previous value of X

**X\u00f8**
Value of X in previous iteration

**Y**
Geometric ordinate of first profile in a channel (cm)

**YOZ0**
$y_0z_0$, product of profile ordinates at the throat; one-fourth of the geometric throat area in a channel (cm²)

**Z**
Geometric ordinate of second profile in a channel (cm)

4.72 **Subroutine AXFIT**

**AG**
Geometric area ratio

**UPD\u00f8WN**
Indicator; -1.0 upstream of throat, 1.0 downstream

4.73 **Subroutine AXSECT**

**I(J)**
List of species for which cross section data are wanted (common array)

**II**
Index of first species in species pair

**IV**
Location of species II in the I array

**JJ**
Index of second species in species pair

**JV**
Location of species JJ in the I array
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>Running index for DØ 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>AM1</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>Symbol</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>MA_l in low Mach number region of integration from $\phi_0$ 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 $d^2M/dx^2$</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 $\phi$ in the current step</td>
</tr>
<tr>
<td>Symbol</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 Lth 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', \text{ eq. I(179)})</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^2W/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>
HCΩN $R_0T_0/W_0$, factor for converting nondimensional enthalpy into cal/g

HE Free-stream specific enthalpy (cal/g)

HF Form factor, $c^*/\theta$

HF1 HF + 1

HREF Reference enthalpy

HTR Incompressible form factor, $H_{inc}$

HTRP1 HTR + 1

HW Enthalpy of the gas at the wall temperature

HO Specific stagnation enthalpy (cal/g)

I $D\phi$ index

ITYPE Index for type of nozzle geometry
ITYPE = 1 Two-dimensional nozzle
ITYPE = 2 Axisymmetric nozzle
ITYPE = 3 Rectangular channel

J $D\phi$ index

K $D\phi$ index over the cold species

L $D\phi$ index running over the nozzle or channel profiles

LPR Index, equal to LPRIME(L)

LPRIME(L) Equal to 2 for $L = 1$, equal to 1 for $L = 2$

$\phi_{LDMF}$ Logical flag; "true" if no mole fraction has changed by 0.01 since the last flow point at which transport properties were calculated
$\phi_{Mega}$  $\omega$, viscosity-temperature exponent in eq. I (191)

$\phi_{M1}$  $\text{Max} \ (1 - \omega, 0)$

$\phi_{NE}$  $1$

$\phi_{Rdin}(L)$  $\bar{e}$ for the boundary layer on the Lth profile, eq. I(171)

$\phi_{Rd1}$  A value of $\bar{e}$ used in the numerical integration from $x_0$ to the first flow point

$\phi_{Rd2}$  See $\phi_{Rd1}$

P  Nondimensional pressure, $p/p_0$

PRPWR  $(\text{PRREF})^{0.56}$

PRW  Prandtl number at the wall temperature

R  Ratio of local nozzle radius to throat radius for axisymmetric nozzles

RA  Intermediate quantity in curvefit to Reynolds analogy factor

RAF  Reynolds analogy factor $R_A$ in eq. I(195)

RB  Intermediate quantity in curvefit to Reynolds analogy factor

RER  Reynolds number based on $R_0$

RH  Nondimensional density

RMR  $\rho_l^\mu$ ratio, $\rho_w^\mu/\rho_e^\mu$

RWMW  $\rho_w$ ;

RI  Radius of a profile in the numerical integration from $x_0$ to the first flow point
R2

SIGH

Hypersonic parameter σ

SQT

Intermediate quantity in transverse curvature correction, eq. 1(187)

SWFAC

\((S_w + 1) [7 + 3.4 (S_w + 1)]\)

SWFAC2

\(1 + 4.5 (S_w + 1)^{0.9}\)

SWP1

\(S_w + 1\)

T

Nondimensional temperature, \(T/T_0\)

TE

Free stream temperature, T

TEP

Free stream temperature at which transport property calculations were last done

THETA1

Momentum thickness based on eq. I(185), divided by \(R_0\)

TR

\(T^*/T_W\), ratio of reference temperature to wall temperature

TRCALC

Logical flag, .TRUE. if transport properties were calculated at the current flow point

TREF

Reference temperature (°K)

TREFP

Reference temperature at which transport properties were last calculated

TRP(I)

Array equivalenced to common block /TRPRP/, to permit processing these data in a DO loop

TRPSV(I)

Array used to save the transport properties at the free stream temperature

U

Nondimensional velocity, \(u/u_s\) (see US in common block /SS/)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UPRIME</td>
<td>Free stream velocity, ( u ) (cm/sec)</td>
</tr>
<tr>
<td>VISCR</td>
<td>Viscosity at the reference temperature (poise)</td>
</tr>
<tr>
<td>VISCW</td>
<td>Viscosity at the wall temperature (poise)</td>
</tr>
<tr>
<td>VR</td>
<td>( \frac{\text{VISCR}}{\text{VISCW}} )</td>
</tr>
<tr>
<td>WD</td>
<td>Weight factor ( w ) in calculation of ( d \delta^*/dx ); eq. 1(419)</td>
</tr>
<tr>
<td>WW</td>
<td>( e^{-\Delta x/a} ) in eq. 1(217); weighting factor for ( \bar{n}_p ) in the calculation of the averaged correlation parameter, ( \bar{n} )</td>
</tr>
<tr>
<td>X</td>
<td>Axial coordinate (cm)</td>
</tr>
<tr>
<td>XII</td>
<td>Previous value of ( XI )</td>
</tr>
<tr>
<td>XL</td>
<td>Shear parameter ( \ell ), eqs. 1(194)</td>
</tr>
<tr>
<td>XP</td>
<td>Previous value of ( x ) (cm)</td>
</tr>
<tr>
<td>XPXP</td>
<td>( X + XP )</td>
</tr>
<tr>
<td>XSAVE(1)</td>
<td>Temporary storage for species mole fractions ( \text{SAVEC}(1) )</td>
</tr>
<tr>
<td>XSNV</td>
<td>( \min(\bar{n},0) ); used in place of ( \bar{n} ) in calculation of HTR to ensure the use of a nonpositive value</td>
</tr>
<tr>
<td>XTEST(J)</td>
<td>Mole fraction of the ( J )th species used in the last previous calculation of transport properties</td>
</tr>
<tr>
<td>XX</td>
<td>Axial coordinate variable in numerical integration from ( X_0 )</td>
</tr>
<tr>
<td>YZ(L)</td>
<td>Radius of the ( L )th profile (cm)</td>
</tr>
<tr>
<td>ZERO</td>
<td>0.</td>
</tr>
</tbody>
</table>
4.75 Subroutine BICALL

FINAL Logical subroutine argument; .FALSE. when BLAYER is called during an intermediate calculation of the nonequilibrium integration, .TRUE. for the final calculation of conditions at the end of each successful integration step.

GJ(J) Species concentrations in moles/g

J Dφ 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 computations

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 current species pair in the V array

MV Index of last parameter in V array for step L-1 of the computations

4.77 Subroutine COMM

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

CIZ2 $\ln (Z2)$

CIZ3 $\ln (Z3)$

EF $\epsilon_f$ in eq. I(32lc)

ER $\epsilon_r$ in eq. I(32lc)

GJ(J) Concentration ($\gamma_j$) of the Jth species (mole/g)

I DØ index over the reactions

ICHG Equal to 1 for a neutral species, equal to 2 for a charged (ionic) species

ICOUNT Counter for number of times CØMM has been called in a run

ICYCLE Interval in ICOUNT at which ISW5B is set equal to 1 to provide diagnostic dumps in subroutine CØMM, EXACT, RNKT, and PRTA

IPAI Index of atomic species produced in current reaction for KTF(I) = 4

J DØ 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 \( T_e \).

LLOGC  Logical indicator for method of calculation of \( \text{PCHI}(I) \); for LLOGC = .FALSE., \( \text{PCHI}(I) \) is computed as \( P_i \gamma_i \); for LLOGC = .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.

QELAS  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, \( T_e (\text{oK}) \)
| TEQ   | $T_e^3$ |
| T6    | $\sum_{j=1}^n \nu_{ij} \ln \gamma_j$ |
| T7    | $\Sigma \gamma$ over the third-body species for a reaction |
| VEJ   | Collision frequency factor $\nu_{eij}$ in the calculation of energy transfer to the electron gas by elastic collisions |
| VIC   | Electron-ion elastic collision frequency $(\nu_{eij})$ divided by the ionic species concentration $\gamma_j$ |
| VIC1  | $(8/3)\sqrt{\pi/m_e} e^4N_0/(2k)^{3/2}$, 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 |
| VIC2  | $\frac{k^3}{(\pi e^6)}$ |
| VNC   | $(4/3) \sqrt{8k/(\pi m_e)} N_0$ |
| VNCC  | Electron-neutral collision frequency $(\nu_{ej})$ divided by the neutral species concentration $(\gamma_j)$ |
| Y     | Radius of first channel profile, cm |
| Z     | Radius of second channel profile, cm |
| Z1    | $\ln u$, natural logarithm of flow velocity in cm/sec |
| Z2    | $R_T$, where $T'$ = $T$ for $KR = 1$ and $T' = T_e$ for $KR = 2$ (cm$^3$ atm/mole) |
| Z2F   | $R_{oT_0}$ (cm$^3$ atm/mole) |
| Z3    | $\rho R_{0T''}$ (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 \(T/T_0\)

**DELI1(L)**
\(S_{11}^*\) for the \(L\)th profile (eq. I(424))

**DELI2(L)**
\(S_{12}^*\) for the \(L\)th profile (eq. I(424))

**DELO1(L)**
\(S_{01}^*\) for the \(L\)th profile (eq. I(424))

**DELO2(L)**
\(S_{02}^*\) for the \(L\)th profile (eq. I(424))

**DEN**
Denominator in eq. I(424)

**HJE**
Temporary storage for nondimensional enthalpy of the electron gas \((H_e/R_0T_0)\)

**I**
\(D\phi\) 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**
\(D\phi\) index over profiles

**SENTE**
Temporary storage for nondimensional entropy of the electron gas \((S_e^0/R_0)\)
4.80 **Subroutine DSMSOL**

A Matrix initially containing the coefficients and constants for a system of linear equations to be solved:

\[
\begin{align*}
    a_{11}x_1 + a_{12}x_2 + \ldots + a_{1i}x_i &= b_1 \\
    a_{21}x_1 + a_{22}x_2 + \ldots + a_{2i}x_i &= b_2 \\
    \vdots &\quad \vdots \\
    a_{ii}x_1 + a_{i2}x_2 + \ldots + a_{ii}x_i &= b_i
\end{align*}
\]

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

LN1 Column index (second index) in close-packed matrix

M1 \( I + 1 \)
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
**IP** Control for printing the time; zero value suppressed the print

4.85 Subroutine **EPART**

**EF** Energy gained by the electron gas in the forward direction of a reaction (cal/mole)

**ER** Negative of the energy gained by the electron gas in the reverse reaction (cal/mole)

**EO** Reaction energy to be partitioned between radiative losses and heating of the electron gas, \( E_0 \) (cal/mole)

**IR** Reaction index in the list of reactions for the current gas model

**IT** Indicator (ITR) of the rule for partitioning the reaction energy in the current reaction

**QF** Energy lost by radiation in the forward reaction (cal/mole)

**QR** Negative of energy lost by radiation in the reverse reaction (cal/mole)

**RO** Universal gas constant, cal/mole-°K

**TE** Electron temperature (°K)

**THR** \( 3R_0/2 \), where \( R_0 \) is the universal gas constant in cal/mole-°K

4.86 Subroutine **ECALC**

**CTSAVE** Variable used to save and restore the initial value of the nondimensional temperature CT

**I** DØ index

**IL** I + ISC
If ISC
Iteration counter for Newton-Raphson solution of equations I(233)

Pressure (atm)

Routine name for calling DUMP routine

Temperature (°K)

Newton-Raphson correction factor \((1 + \frac{h}{\gamma})\) to mole fractions, eq. I(235)

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]

4.87 *subroutine EXACT*

AAA(L,M) Array of coefficients and constants for system of linear equations

\[ D\phi \text{ index} \]

\[ D\phi \text{ index} \]

\[ I + ISC \]

\[ \text{Iteration counter for Newton-Raphson solution of equations I(233)} \]

\[ \text{Pressure (atm)} \]

\[ \text{Routine name for calling DUMP routine} \]

\[ \text{Temperature (°K)} \]

\[ \text{Newton-Raphson correction factor (1 + h/\gamma)} \text{ to mole fractions, eq. I(235)} \]
\[
\begin{align*}
S_I &= -1 + \nu_i^* + \frac{H_i}{R_0 T} - \sum_{j=1}^{C} \nu_{i-c-j} \frac{H_j}{R_0 T} \\
1 &= \left(\frac{u}{u_s}\right)^2/W_0
\end{align*}
\]

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 \(I\)th 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 FINDXC

~~MBL~~

Index (1 or 2) of the profile whose ordinate varies most rapidly downstream of the throat in a channel

~~N~~

Iteration counter
<table>
<thead>
<tr>
<th><strong>Variable</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNAME</td>
<td>Routine name for call to DUMP routine</td>
</tr>
<tr>
<td>UDOWN</td>
<td>Indicator; -1.0 if upstream solution is desired, 1.0 if downstream solution is desired</td>
</tr>
<tr>
<td>UPPER</td>
<td>Logical indicator; .TRUE. if a positive ERA has been found at some trial value of X</td>
</tr>
<tr>
<td>V</td>
<td>Trial value of</td>
</tr>
<tr>
<td>VL</td>
<td>Trial value of</td>
</tr>
<tr>
<td>VØ</td>
<td>Trial value of</td>
</tr>
<tr>
<td>VU</td>
<td>Trial value of</td>
</tr>
<tr>
<td>X</td>
<td>Final solution for</td>
</tr>
<tr>
<td>Z(I)</td>
<td>Ordinate of the Ith profile in a channel</td>
</tr>
</tbody>
</table>

4.90 **Subroutine FROOT**

<table>
<thead>
<tr>
<th><strong>Variable</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Common variable, used here for the geometric area ratio</td>
</tr>
<tr>
<td>AS</td>
<td>Sound speed (cm/sec)</td>
</tr>
<tr>
<td>BLVAR(I)</td>
<td>Array equivalenced to the contents of common block /BLOUT/</td>
</tr>
<tr>
<td>BLVARØ(I)</td>
<td>BLVAR values at preceding flow point</td>
</tr>
<tr>
<td>BLVARS(I)</td>
<td>BLVAR values at the current regular flow point; used in interpolating at a model point</td>
</tr>
<tr>
<td>CNMMA</td>
<td>Logarithm of the molecular weight in the reservoir</td>
</tr>
<tr>
<td>CTSAVE</td>
<td>Nondimensional temperature at the regular flow point following a model point</td>
</tr>
</tbody>
</table>
CTSTAR  The nondimensional temperature (CT) value at
the flow point preceding the throat, in solutions including the boundary layer

CT1   Second most recent trial value of CT in the
iteration to find the conditions at a model
point

CT2   Most recent trial value of CT in iteration
to find the conditions at a model point

CXSAVE  X coordinate of the regular flow point fol-
lowing a model point

DEBLP(I)  For I = 1, 2, the nondimensional boundary
layer displacement thickness for the Ith pro-
file at the preceding flow point

DEBLBP(I)  For I = 1, 2, the nondimensional boundary
layer displacement thickness for the Ith pro-
file at the point before the preceding flow
point

DELT   Decrement in nondimensional temperature CT in
calculation of frozen throat conditions

DELTSV   Variable used to save the input value of DELT1

DELTv   Current decrement in nondimensional tempera-
ture CT

ERRX   Position error in iteration to determine the
conditions at a model point

FLAG   Indicator for upstream and downstream regions,
for use by the geometry subroutines. Negative
value indicates an upstream position, positive
value a downstream position
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1, F2, F3</td>
<td>Values of nondimensional mass flux in iteration to determine the frozen throat conditions</td>
</tr>
<tr>
<td>GJ(J)</td>
<td>Species concentrations, mole/g</td>
</tr>
<tr>
<td>I</td>
<td>DØ index</td>
</tr>
<tr>
<td>IPASS</td>
<td>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</td>
</tr>
<tr>
<td>IRSTRT</td>
<td>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</td>
</tr>
<tr>
<td>ISØLN</td>
<td>Indicator for type of flow solution:</td>
</tr>
<tr>
<td>ISØLN = 1</td>
<td>Frozen solution</td>
</tr>
<tr>
<td>ISØLN = 2</td>
<td>Equilibrium solution</td>
</tr>
<tr>
<td>ISØNIC</td>
<td>Indicator for region of flow solution:</td>
</tr>
<tr>
<td>ISØNIC = -1</td>
<td>Upstream region</td>
</tr>
<tr>
<td>ISØNIC = 0</td>
<td>Sonic point</td>
</tr>
<tr>
<td>ISØNIC = 1</td>
<td>Downstream region</td>
</tr>
<tr>
<td>ISWITCH</td>
<td>Counter for number of flow points beyond the throat</td>
</tr>
<tr>
<td>ITS</td>
<td>Index of the next specified test section diameter (TSDIAM)</td>
</tr>
<tr>
<td>J</td>
<td>DØ index over gas species</td>
</tr>
<tr>
<td>K</td>
<td>DØ index</td>
</tr>
<tr>
<td>LASTPT</td>
<td>Indicator for last point in flow solution (at CX = CXMAX)</td>
</tr>
<tr>
<td>M</td>
<td>DØ index over channel profiles</td>
</tr>
<tr>
<td>MODELPT</td>
<td>Logical indicator; if .TRUE., a model point is being calculated</td>
</tr>
<tr>
<td>N</td>
<td>DØ index</td>
</tr>
</tbody>
</table>
NMPI  Counter for number of iterations to determine the conditions at a model point

ØNE  1.0

SKIP  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

S1  \[-\frac{1}{W_0} \sum_{j=1}^{n} X_{j0} \ln X_{j0}\]

S2  \[-(\ln p_0)/W_0\]

TSTEP C  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

XLAST  Position of preceding flow point (cm)

XMØDEL  Model-point position (cm)

XMØDL  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)
X1  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)

X2  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)

Z  \( F1 - F2 + F3 - F2 \)

4.91  **Subroutine GEOM**

AN  Effective area ratio corresponding to the current trial value of \( \rho \), based on the area ratio-density relation, eq. I(383)

A2  Square of the effective area ratio

ARG  \( 2C(A_e-1)/\alpha \)

DADR  \( \rho_0 dA_e/d\rho \), derivative of the effective area ratio with respect to nondimensional density

DD  Correction in nondimensional density

DELRHØ  Absolute value of the difference between \( \rho/\rho_0 \) and the nondimensional density at the throat, based on approximate solution near throat

DELSTP  Displacement thickness at the throat (cm)

D2IA  \( [ d^2(\ln A_e)/dx^2 ] \), the second derivative of the logarithm of the effective area ratio at the throat

J  \( \psi \) index over the profiles in a channel

N  Iteration counter

NSUJ  Number of upstream sections in the curvefit to a profile
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>φMZ</td>
<td>( l - z )</td>
</tr>
<tr>
<td>φPHA</td>
<td>( 1 + \alpha/2 ), where ( \alpha ) is the exponent in the density-area ratio relation I(383)</td>
</tr>
<tr>
<td>RNAME</td>
<td>Routine name for call to DUMP</td>
</tr>
<tr>
<td>SGN</td>
<td>-1 if ( dA_g/dx ) is positive, +1 if ( dA_g/dx ) is negative</td>
</tr>
<tr>
<td>Y</td>
<td>Consistency check on ( x ) and ( \rho ) values; if positive, ( x ) and ( \rho ) data are inconsistent</td>
</tr>
<tr>
<td>Z</td>
<td>( (\rho/\rho_0)^\alpha )</td>
</tr>
</tbody>
</table>

4.92 Subroutine GEΩMAR

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARATIØ</td>
<td>Geometric area ratio</td>
</tr>
<tr>
<td>DAB</td>
<td>Difference between ( x ) and the ( x )-coordinate of the center of a circular profile section</td>
</tr>
<tr>
<td>DERIVA</td>
<td>( dA_g/dx ), derivative of geometric area ratio ( (\text{cm}^{-1}) )</td>
</tr>
<tr>
<td>DRV(I)</td>
<td>Slope (derivative) of ordinate for ( I )th profile</td>
</tr>
<tr>
<td>DYDX</td>
<td>Slope of ordinate for first profile, after rescaling area</td>
</tr>
<tr>
<td>DZDX</td>
<td>Slope of ordinate for second profile, after rescaling area</td>
</tr>
<tr>
<td>I</td>
<td>Index of two profiles</td>
</tr>
<tr>
<td>IENTRY</td>
<td>Index of entry point: 1 - GEΩMAR; 2 - GMAR; 3 - GMAR2; 4 - GMAR3</td>
</tr>
<tr>
<td>ISJ</td>
<td>Shape index for a profile section</td>
</tr>
</tbody>
</table>
J  Index for the sections in a profile
K  Index for the profile section in which X lies
NSML 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)
RAD(I) Ordinate of the Ith profile (cm)
RAT Ratio of a profile ordinate to the ordinate at the throat
SN(I) -1.0 for I = 1, + 1.0 for I = 2
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 Subroutine INGAS
CAPQ(K) Q_k, number of gram-atoms of the kth chemical element per mole of the cold gas
CCI(I) Intermediate variable in the calculation of the molecular weights of species from atomic weights of elements
Species index

IL  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 triatomic 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 $T_0(K)$

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 \( \alpha^\alpha \) 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)

\( \sqrt{T} \) is the translational thermal conductivity in milliwatts/cm-°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 \( NJ > NI \)
$N1 \quad N + 1$

$ZKINT \quad ZKINT \sqrt{T}$ is the internal thermal conductivity in milliwatt/cm-°K

4.97 **Subroutine LIST**

$CAPQ(I) \quad Q_k$, number of gram-atoms of the kth chemical element per mole of the cold gas

$CØM1(I,J) \quad$ Hollerith array for output (see DATA statement)

$CØM2(I,J) \quad$ Hollerith array for c.tput (see DATA statement)

$CXMAXI \quad$ Distance beyond throat at which case will be terminated (inches)

$I \quad$ DØ index

$J \quad$ DØ index

$K \quad$ DØ index

$KDQL \quad$ KDIM + 1

$NWXI \quad$ Number of specified distances WXI from the leading edge of a wedge model

$SBJ(J) \quad b_j$, eqs. I(51)

$SHJAP(J) \quad$ Formation enthalpy of the Jth species, cal/mole

$THEVP(J) \quad$ Characteristic vibrational temperature for the Jth species, °K

$WØRD(I,J) \quad$ 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
**RNAME**
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/cm²

**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

**EILN(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 $\epsilon = \rho_1/\rho_2$ across normal shock

**EPSOLD**
Previous estimate of $\epsilon$

**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>Definition</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>ETO</td>
<td>Elapsed time since the beginning of the run when <code>MODEL</code> 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 (<code>RT ø/ø</code>) 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/\bar{u}_g)^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 <code>.TRUE.</code>., 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>HRATIØ</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>Dφ index</td>
</tr>
<tr>
<td>ICOUNT</td>
<td>Iteration counter</td>
</tr>
<tr>
<td>ISOLN</td>
<td>Index specifying the two types of normal-shock solution (equilibrium ISOLN = 1, frozen ISOLN = 2)</td>
</tr>
<tr>
<td>ISSNP1</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>Dφ index</td>
</tr>
<tr>
<td>L</td>
<td>Model-shape index</td>
</tr>
<tr>
<td>LIMSCI</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 Dφ loop for output of mole fractions</td>
</tr>
<tr>
<td>LIM2</td>
<td>Upper limit of implied Dφ loop for output of mole fractions</td>
</tr>
<tr>
<td>LMSC</td>
<td>Number of data items from the SCOUT 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>Dφ index</td>
</tr>
<tr>
<td>NEL</td>
<td>Number of chemical elements in a species</td>
</tr>
</tbody>
</table>
PC  
4 \( \rho_1^2 u_1^2 \), atm-\( g/cm^3 \)

PRW  
Prandtl number at the model wall temperature

PS  
Stagnation pressure, atm

PT  
\( p_1 + \rho_1 u_1^2 \), 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 \( L = 2 \)).

QF  
\( p_1 T_2 / T_1 \)

QS  
Factor used in calculating QFRE and QFRF

QSRI  
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\O STR  
Density at reference temperature in flat-plate calculation

R1  
Density ahead of the shock, g/cm\(^3 \)

R1U12  
\( \rho_1 u_1^2 \), 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 ERRHI, \( ^\circ K \)

TTITLE2

Hollerith data used in output

TLØ

\( T_2 \) value corresponding to enthalpy error ERRŁØ, \( ^\circ K \)

TSTAR

Reference temperature in flat-plate calculation, \( ^\circ K \)

T1

Temperature ahead of the shock, \( ^\circ K \)

T2

Temperature behind the shock, \( ^\circ K \)

U1

Velocity ahead of the shock, cm/sec

VGP(L)

Velocity gradient parameter \( (R_s/u_1)\frac{d\theta_p}{dx} \)
(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

-295-
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 \(1+n_{i}^l\) to the mole fractions, eq. I(260)
ZC Correction factor \(1+h_{C+1}^l\) 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 Temporarily storage location used in interchanging rows of \(\beta_{ij}\) matrix during determination of matrix rank

-296-
ADCH Absolute value of a $\delta \chi_i$ value

ALGT $\ln (10000)$

BLNEAR(M) Array equivalenced 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) $\frac{1}{(\gamma_k')} \sum_{j=1}^{c} \delta c_j a_{jk}$ in eq. I(346c)

DCHMAX Largest $|\delta \chi_i|$ for any reaction

DCHMIN Smallest $|\delta \chi_i|$ 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

DELIB(L) Nondimensional boundary layer displacement thickness for the Lth profile, $\delta u/R_0$, at the start of an integration step
DELT  Nondimensional temperature interval in the perturbation solution, $\Delta T / T_0$

DELTSV  Input value for DELT

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  DØ 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 IFAIL 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$\Omega$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 |
| **JA** | Index used in calculating the rank of $\beta_{ij}$ |
| **L** | $D\phi$ index |
| **M** | Index used in calculating the rank of $\beta_{ij}$ |
Index used in calculating the rank of $\beta_{ij}$

Number of reactions suppressed when the decrease of a minor species is found to control the integration step

1.0

Energy transfer to the electron gas at the preceding flow point, cal/cm$^3$ sec

Ratio of the smallest and largest $|\gamma_i|$ values

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

Subroutine name for call to DUMP routine

Factor by which the step size $\Delta x$ is divided when an integration step fails and has to be restarted

Change in the concentration $\gamma_j$ of the $J$th species in an integration step

10000

Electron temperature ($^\circ$K)

Gas temperature ($^\circ$K)

Temporary storage for TPRINT

Position of next model point, cm

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 XMMODEL

ZZZ Temporary storage

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 DØ index over species

J DØ index over species

T1,T2,T3 Nondimensional temperature values in iteration to determine the equilibrium sonic flow conditions

Z F1 - F2 + F3 - F2

4.103 Subroutine NRMAX

4.104 Subroutine ØUT1

ANAM Hollerith word "ARAT"

ARAT Geometric area ratio

ASØLN(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 R₀
DARAT  Derivative of the geometric area ratio
DEL    Electron density, electrons/cm³
DELSTP(L)  Displacement thickness on the Lth profile, cm
DGJ(I)  d γ_i/dx, derivatives of species concentrations, mole/g-cm
DΛGDX(I)  d lnγ_i/dx for the Ith species
DNAM  Hollerith work "DIAM"
EMF    Mole fraction of the electrons
FAREA  Effective cross-sectional area of the flow at the throat, cm²
FLX    Mass flux, lbm/ft²·sec
FVψ2(I,J)  Array used for output of SN and XSN
GAMMA  Ratio of specific heats, γ = c_p/c_v
GJ(I)  Species concentrations, mole/g
HOUT  Enthalpy, Btu/lb
HW    Enthalpy at the nozzle wall, Btu/lb
HO    Stagnation enthalpy, Btu/lb
I     DØ index
IBL    Index such that FVØUT(IBL) is the boundary layer displacement thickness on the nozzle wall or on the broad face of a channel
IFLAG  Indicator for entry points to subroutine
        IFLAG = 1  Entry ØUT1
        IFLAG = 2  Entry ØUT2(ISØIN)
II    Index 7 + 10*I
INEQP1  INEQ + 1
ISQLN  Indicator for type of solution
      ISQLN = 1  Frozen
      ISQLN = 2  Equilibrium
      ISQLN = 3  Nonequilibrium
ISSPL1  (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
PZ     3.1415927
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RØUT</td>
<td>Density, lb/ft³</td>
</tr>
<tr>
<td>SØUT</td>
<td>Entropy, Btu/lbm-°R</td>
</tr>
<tr>
<td>STEP(I)</td>
<td>Hollerith array (see DATA statement)</td>
</tr>
<tr>
<td>STFAC</td>
<td>( \frac{1}{\rho u(h_0-h_w)} ), (Btu/ft²·sec)⁻¹</td>
</tr>
<tr>
<td>TFLyw</td>
<td>Total mass flow, lb/sec</td>
</tr>
<tr>
<td>TNNAM(I)</td>
<td>Hollerith array of variable names (see DATA statement)</td>
</tr>
<tr>
<td>TØUT</td>
<td>Temperature, °K</td>
</tr>
<tr>
<td>TYPsln(I,J)</td>
<td>Hollerith array used to indicate whether nonequilibrium solution is based on perturbation technique (INEQ = 0) or numerical integration (INEQ = 1)</td>
</tr>
<tr>
<td>UØUT</td>
<td>Velocity, ft/sec</td>
</tr>
<tr>
<td>VARNM(I)</td>
<td>Array of Hollerith names for output variables (see DATA statement)</td>
</tr>
<tr>
<td>VARN2(I)</td>
<td>Array of Hollerith names for N, XSN</td>
</tr>
<tr>
<td>WNAM</td>
<td>Hollerith word &quot;WIDTH&quot;</td>
</tr>
<tr>
<td>Y</td>
<td>Distance from nozzle axis to nozzle surface (or to first profile in the case of a channel)</td>
</tr>
<tr>
<td>Z</td>
<td>Distance from axis to second profile of a channel</td>
</tr>
</tbody>
</table>

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 PIøMEG
I
  Index of entries in cross section table
PIøMEG
  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

4.108 Subroutine PRTC
S4
  \[ \frac{1}{R_0} \sum_{j=1}^{n} \gamma_j \theta_j^o \]

4.109 Subroutine PUTQIN
AS
  Speed of sound, cm/sec

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. \( \phi_M(1) = \Omega^{(1,1)} \), \( \phi_M(2) = \Omega^{(2,2)} \), \( \phi_M(3) = B*\Omega^{(1,1)} \)

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 QCQU

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

\( \phi_M \) 0.80\( \phi_c \) (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 QUE

A  Coefficient A in eq. I(104) for the exchange cross section
B  Coefficient B in eq. I(104) for the exchange cross section

$\phi M(K)$  Contribution to cross sections from option;
$\phi M(1) = \lambda (1,1)\phi M(2) = \lambda (2,2)$
$\phi M(3) = B*\lambda (1,1)$

$V(M)$  Input parameters $VW$ for option

$Y$  Quantity $A - 3/2 \lambda \ln (T/M)$

$ZM$  Molecular weight

4.112 Subroutine QEXP

A  Coefficient $A/k$ in $^0K$ for the exponential potential

ALPHA  Quantity $\alpha$

N1  Position of first entry in tabulated collision integrals for the exponential potential

$\phi M(K)$  Computed cross section values (see subroutine QEX)

RHO  Characteristic length $\rho$ for the exponential potential in $\hat{R}$

$V(M)$  Input parameters $VW$ for option
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

EPSLN
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

$\Theta_{M(K)}$
Computed cross section values (see subroutine QEX)
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \text{SIGMA} )</td>
<td>Collision diameter ( \sigma ) for the Lennard–Jones potential, in Å</td>
</tr>
<tr>
<td>( \text{TSTAR} )</td>
<td>Nondimensional temperature, ( kT/\epsilon )</td>
</tr>
<tr>
<td>( \text{V(M)} )</td>
<td>Input parameters ( VV ) for option</td>
</tr>
</tbody>
</table>

### 4.115 Subroutine QMIX

- \( I \) | 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 QREPP

- \( ITL \) | Location in OMEGAL array where data are stored |
- \( \Omega M(K) \) | Computed cross section values (see subroutine QEX) |
| \( \text{VV(M)} \) | 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

\[ \phi M(K) \] Computed cross section values (see subroutine QEX)

\[ VV(M) \] 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

\[ \phi M(K) \] Computed cross section values (see subroutine QEX)

\[ V(M) \] Input parameters VV for option

4.119 Subroutine Q11

\[ LQ1 \] 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 \(^0\text{K}\) beyond which cross sections are set to zero

\[ T1 \] Temperature in \(^0\text{K}\) beyond which cross sections are unchanged

\[ VI(M) \] Input parameters VV for option
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)

4.121 Subroutine Q13

**C**
Constant by which cross section is multiplied

**I**
First index of species pair

**J**
Second index of species pair

**L1**
Type of cross section to be computed

**L2**
Type of cross section to be used in computation

**IΩ**
Location of species pair in IQ, JQ list

**IΩ1**
Location in IQ, JQ list of first species pair to be used
LQ2    Location in IQ, JQ list of last species pair to be used
VV(M)  Input parameters VV for option

4.122 Subroutine Q14

C2     Constant applied to \( \bar{\Omega} \) \((2,2)\)
C3     Constant applied to \( B \times \bar{\Omega} \) \((1,1)\)
I      First index of species pair
J      Second index of species pair
LQ     Location of species pair in IQ, JQ list
LQ1    Location in IQ, JQ list of the first species pair to be used with the set of multiplying factors
LQ2    Location in IQ, JQ list of last species pair to be used with the set of multiplying factors
VV(M)  Input parameters VV for option

4.123 Subroutine RADIUS

AG     Geometric area ratio
AGJ    Factor proportional to \( r^2 \) in eq. I(171)
ITYPE  Indicator for type of nozzle geometry
         ITYPE = 1 Two-dimensional nozzle
         ITYPE = 2 Axisymmetric nozzle
         ITYPE = 3 Channel
L      Index of profiles (1 for a nozzle, 1 or 2 for a channel)
M NPR(L)
NPR(L) 2 for L = 1; 1 for L = 2
R Distance from axis to profile
RATIØ R/R0
X Axial coordinate in nozzle or channel, cm
YZ(L) Distance from axis to Lth profile in a channel

4.124 Subroutine READ

AL Number of atoms of the Kth element per molecule of the Ith species, as obtained from the SPRP array
ASYM(IZ) Hollerith chemical symbol for the element with atomic number IZ
CAPQ(I) Number of gram atoms of the Ith chemical element per molecule of the cold gas
CP(I,J) Array for addressing the channel data in common block /CHAN/
CSMW Molecular weight of one of the cold species
DAL Error in rounding A to an integer LP
DUM(I) Array for zeroing common block /TNCE/
ELNAME(K) Hollerith name for the Kth element in one of the cold species
EPRP(I,J) Array for addressing the element data in common block /ELEM/
EXPMP: Exponent used in computing the factor (FACMP) used to generate a geometric sequence of distances \( x \) at which model calculations will be done.

FNMP: Number of model points in geometric sequence, minus 1.

GPRP(I,J): Array for addressing the gas model data in common block /MIXT/.

HWIN\(\Phi\): Estimate of stagnation enthalpy based on Wino-vich equation, Btu/lb.

I: D\(\Phi\) index.

IES: Index of an element in a species, in the master list of elements.

IGMV: Number of electronic levels in a species.

IGS: Input value of IGAS.

II: Index used in place of subscripted integer variables.

IPRFL: Index used in place of subscripted integer variable, NPR\(\Phi FL(J)\).

ISH: A value of ISHAPE(K,J).

ISKUR: Index of a third-body species in the master list of species.

IUPDI: Initial value of IUPD.

IZ: Atomic number of an element defined in the input using EEPRP.

J: D\(\Phi\) index.

JDP1: JDIM + 1.
JJ  Index used in place of a subscripted integer variable, JCS(I)

JPS  Index of a species on the product side of a reaction, in the master list of species

JRS  Index of a species on the reactant side of a reaction, in the master list of species

K    DØ index

KK   Index of an element in a cold species, in the master list of elements

L    DØ index

LL   Index used in obtaining the PARAM array from the data stored in the ZPRP array

LP   Integer obtained by rounding AL

NATØM Number of atoms of an element in a molecule of a species

NELS Number of elements in a species

NK   Number of third body species for a reaction

NOZZLE Index of the nozzle profile in the precoded geometry fits

NPSS Number of product species for a reaction

NRSS Number of reactant species for a reaction

NSECTD(J) Number of downstream sections in the profile curvfit for the Jth profile of a channel or nozzle

NSECTJ Index used in place of dimensioned integer variable, NSECT(J)
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/
WØRD(I,J) Hollerith data for output of the words "two-dimensional" and "axisymmetric"
WØRD2(I) Hollerith data for output of the words "gap" and "diam"
ZPRP(I,J)  Array for addressing the geometry data in common block /NZZZ/

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 RZSTMP

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

CONST  $R_0\left(\frac{m_s}{p_0}\right)^2$, in which $R_0$ is the universal gas constant, $m_s$ the known sonic mass flux, and $p_0$ the reservoir pressure, all expressed in cgs units

CTAPP  Previous estimate of reservoir temperature, °K

CSTAVE(I)  For $I = 1$ to 3, successive estimates of the reservoir temperature, the value for $I = 1$ being the most recent, °K

CTSV  Temporary storage for the reservoir temperature estimate used in the current iteration, °K

DCT  The difference between the two most recent estimates of the reservoir temperature, °K
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCTAP</td>
<td>Absolute value of the difference between the current and improved estimates of the reservoir temperature, °K</td>
</tr>
<tr>
<td>DCTAPφ</td>
<td>DCTAP in the previous iteration step, °K</td>
</tr>
<tr>
<td>DMC</td>
<td>Intermediate quantity in quadratic interpolation for the reservoir temperature, °K</td>
</tr>
<tr>
<td>D1,D2</td>
<td>Intermediate quantities in linear interpolation for the reservoir temperature, °K</td>
</tr>
<tr>
<td>F(I)</td>
<td>For I = 1 to 3, successive estimates of the quantity ( W_0 \cdot (SM)^2 ), the value for ( I = 1 ) being the most recent; ( W_0 ) is the molecular weight in the reservoir and ( SM ) the nondimensional sonic mass flux</td>
</tr>
<tr>
<td>H</td>
<td>Calculated value of reservoir enthalpy, cal/g</td>
</tr>
<tr>
<td>HERR</td>
<td>Allowable error in calculated stagnation enthalpy cal/g</td>
</tr>
<tr>
<td>Hφ</td>
<td>Calculated stagnation enthalpy in previous iteration step, cal/g</td>
</tr>
<tr>
<td>I</td>
<td>Dφ index</td>
</tr>
<tr>
<td>J</td>
<td>Index for shifting data in CTSAVE and F arrays</td>
</tr>
<tr>
<td>KS</td>
<td>Indicator for singularity of AM matrix</td>
</tr>
<tr>
<td>N</td>
<td>Iteration counter for main iteration</td>
</tr>
<tr>
<td>N2</td>
<td>Iteration counter for temperature calculation in enthalpy option</td>
</tr>
<tr>
<td>N3</td>
<td>Iteration counter for pressure calculation in enthalpy option</td>
</tr>
</tbody>
</table>
PCGS  Reservoir pressure, dyne/cm²
PERR  Allowable relative error in reservoir pressure
PIAST  Previous estimate of reservoir pressure, atm
RGAS  Universal gas constant in cgs units
RNAME  Hollerith name for calling DUMP routine
SMERR  Allowable error in sonic mass flux, g/cm² sec
SMP  Calculated sonic mass flux, g/cm² sec
TERR  Allowable relative error in reservoir temperature
TLAST  Previous estimate of reservoir temperature, °C

4.127 Subroutine RNKT

\[ \gamma_j \] 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 \( J \)th dependent variable at the start of the integration step

F2(J)  \( f_2 \), derivative of the \( J \)th dependent variable at \( x_2, y_2 \) (eq. I(394a))

F3(J)  \( f_3 \), derivative of the \( J \)th dependent variable at \( x_3, y_3 \) (eq. I(394b))
\( f_4 \), derivative of the Jth dependent variable at \( x_4 \), \( y_4 \) where \( y_4 \) is given by I(394c) or I(405)

\( \text{GJK}(J,K) \)
Array equivalence to \( \text{GJ1}(J), \text{GJ2}(J), \text{GJ3}(J), \text{GJ4}(J) \)

\( \text{GJ1}(J) \)
y\(_1\), the Jth dependent variable at the start of the integration step

\( \text{GJ2}(J) \)
y\(_2\), the Jth dependent variable at \( x_2 \) (eq. I(394a))

\( \text{GJ3}(J) \)
y\(_3\), the Jth dependent variable at \( x_3 \) (eq. I(394b))

\( \text{GJ4}(J) \)
y\(_4\), the Jth dependent variable at \( x_4 \) (eq. I(394c) or I(405))

\( \text{IF\&R} \)
Indicator for type of formula used to compute \( \text{GJ4}(J) \); \( \text{IF\&R}=1 \) for I(394c), \( \text{IF\&R}=2 \) for eq. I(405)

\( \text{ISTMNT} \)
Indicator for locations in subroutine at which diagnostic dumps are written (see listing)

\( J \)
Index of dependent variables:

\[
\begin{align*}
J &= 1 \text{ to } n & \gamma_j \\
J &= n + 1 & T_j \\
J &= n + 2 & T_e \\
J &= n + 3 & h_0
\end{align*}
\]

\( K \)
D\& index

\( \text{LIM} \)
Number of dependent integration variables

\( P(J) \)
Parameter \((-P \Delta x)\) for the Jth dependent variable, eq. I(402)

\( \text{QDPEB} \)
Electron energy transfer (QDPE) at the start of the integration step, cal/cm\(^3\) sec

\( \text{SDGJ}(J) \)
Change in \( \gamma_j \) over an integration step, mole/g
SDQ(J)  Change in Jth dependent variable over the integration step
S1    Temporary storage in computation of I(398a) for small P Δx
S2    Temporary storage in computation of I(398b) for small P Δx
S3    Temporary storage in computation of I(398c) for small P Δx
TE    Logical indicator, .TRUE. for chemical non-equilibrium models, .FALSE. for electronic nonequilibrium models
T1    Temporary storage in computation of I(398a) for small P Δx
T2    Temporary storage in computation of I(398b) for small P Δx
T3    Temporary storage in computation of I(398c) for small P Δx
XX    \[ \log_{10}( |f_3 - f_2|) \]
X1    \( F_1 \) (eq. I(398a))
X2    \( F_2 \) (eq. I(398b))
X3    \( F_3 \) (eq. I(398c))
X4    \( f_2 - f_1 \)
Y(K)  Array equivalenced to Y1, Y2, Y3, Y4
XY    \[ \log_{10}( |f_2 - f_1|) \]
Y1    \( f_1 + p_{Y1} \) in I(401)
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y2</td>
<td>( f_2 + Py_2 ) in I(401)</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 \( - \left[ \frac{M^2 + 2}{M^4} + \gamma \sin^2 \alpha \right] \), where \( M \) is the free-stream Mach number, \( \gamma \) the specific heat ratio, and \( \alpha \) the body angle

**BETA**

The quantity \( ALP \cdot \sin^2 \sigma + C \)

**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**

\( -\cos^2 \alpha /M^4 \)

**DELR**

The body angle \( \alpha \); inclination of the wedge surface to the direction of free-stream motion (degrees)

**DELR**

\( \alpha \) in radians
**DIS**  \[ \sqrt{(\text{ALP})^2 - 4 \cdot \text{BETA}} \]

**FL(1)** 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, \( T_{stag} / T_1 \)

**FP** 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

A(I) Matrix of coefficients for the system of equations to be solved, stored columnwise with no gaps

B(I) Constant terms for the system of equations to be solved; these data are replaced by the solution values

BIGA Pivotal (largest) element in a column

I Row index

IA Index of matrix A in back solution

IB Index of recomputed array B in back solution, containing solution values

IC Index of array B in back solution

IJ Index in search for pivotal element in column

IMAX Index of pivotal element in column

IQS Index in elimination of variables

IT Index used for several purposes

IX DØ index in elimination of variables

IXJ Index in elimination of variables

IXJX Index in elimination of variables

I1,I2 Indices in interchange of rows

J Column index

JJ Index of the first element in a column of the reduced matrix
JJX  Index in elimination of variables
JX  DΩ index in elimination of variables
JY  J + 1
K  DΩ 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
TΩL  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  DΩ index; also value of IQ(LQ); 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(IQ)
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
IQ1 Index of the first species pair (in the IQ,JQ arrays) to which the Lth step of the edited cross section computation is applied
IQ2 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

TRP(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

SZ1 Temporary storage for a value required in the entropy calculation for a linear triatomic species

S1 See eq. I(56:)

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 $(S2)^2$

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

Z1-Z9 Storage for intermediate values in calculations of the vibrational contributions to species properties
4.132 Subroutine THRØAT

AG
Geometric area ratio

I
DØ index

ØNE
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

CPTYØ
Nondimensional frozen specific heat at constant pressure for the mixture, $W \frac{c_{pf}}{R_0}$

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
$\sqrt{T TAB}$

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 I\textsuperscript{th} 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 IT\textsuperscript{th} column of output table
A2 \( A^2 \)
Angle-of-attack parameter $\Gamma$ (eq. I(481c) or I(501))

Heat transfer coefficients, $C_H$

Coefficient $0.332 A \varepsilon T N^2 t$ for calculating $C_H$ from $S^*$

Coefficient $0.332 A \varepsilon$

Single-precision value equal to $\Gamma$

Boundary layer displacement thickness at the ITth point along the wedge

Coefficient $\gamma \varepsilon \kappa^3/8 A^5 \varepsilon$

Quantity $DFAC \cdot t/(\tau N)^4$

Parameter $N = M \sqrt{C/Re}$

Quantity $\varepsilon = (\gamma - 1)/(\gamma + 1)$

$\varepsilon^2$

Quantity $1 + \text{erf}(S \sin \alpha)$ used in calculating free-molecule limit to heat flux

Quantity $c^{-(S \sin \alpha)^2}$ in free-molecule heat flux calculations
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.

Array used by subroutine SHOCK to communicate its results to WEDGE. See glossary of symbols in SHOCK for definitions of the elements of FL.

Specific-heat ratio, $\gamma$

Double-precision value of GAMMA, for call to SHOCK routine.

Factor used in calculating $\Gamma$; equal to $k/2A^2\varepsilon M$ for the modified Cheng-Kemp theory, and to $\gamma k/2 A^2 \varepsilon$ for the unmodified theory.

$$GFAC = \frac{GFAC}{(\tau N)^2}$$

Coefficient $k/2A^2\varepsilon$ in GFAC.

Mean molecular weight of gas, g/mole.

$$\gamma^2$$

$$\gamma^3$$

Ratio $h_w/h_0$ of enthalpy at the wall to the free-stream stagnation enthalpy.

Gas enthalpy at the wall temperature, cal/g.

Free-stream stagnation enthalpy, cal/g.

Index of angles of attack.

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 DΩ 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

ΩMEGA  Parameter Ω, eq. I(502)

PFAC  Quantity 4 A^4 e^2 M^2 / γ k^2

PFAK  Coefficient PFAC · (τ N)^4 in formula for pressure on wedge

PI  Value 3.1415927

PINF  Free-stream pressure, atm
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRATIOΦ</td>
<td>Ratio of pressure on wedge to free-stream pressure, $p_w/p_1$</td>
</tr>
<tr>
<td>PW(IIT)</td>
<td>Pressure on the wedge at the ITth distance from the leading edge, atm</td>
</tr>
<tr>
<td>QFM(IA)</td>
<td>Free-molecule limit to the heat flux for the IAth angle of attack, Btu/ft²·sec</td>
</tr>
<tr>
<td>QFMCL</td>
<td>Coefficient $0.0885, p_{1a}e^{\sqrt{RT_1/2, \pi , W}}$ in calculation of free-molecule heat flux, eq. I(512)</td>
</tr>
<tr>
<td>QFMCl</td>
<td>Coefficient $S^2 + [\gamma - \frac{1}{2} (\gamma+1) , T_w/T_1]/(\gamma-1)$ in free-molecule heat flux calculation, eq. I(512)</td>
</tr>
<tr>
<td>QFW(IIT)</td>
<td>Heat flux to the wedge at the ITth distance from the leading edge, Btu/ft²·sec</td>
</tr>
<tr>
<td>REPFF</td>
<td>Reynolds number per foot</td>
</tr>
<tr>
<td>RGC</td>
<td>Gas constant $(8.314 \times 10^7$ erg/mole °K)</td>
</tr>
<tr>
<td>RHØINF</td>
<td>Free-stream density, $\rho_1$</td>
</tr>
<tr>
<td>RNAME</td>
<td>Hollerith routine name for call to subroutine DUMP</td>
</tr>
<tr>
<td>RU2</td>
<td>Dynamic pressure $\rho_1 u_1^2$</td>
</tr>
<tr>
<td>S</td>
<td>Free-molecule flow parameter $M\sqrt{\gamma/2}$</td>
</tr>
<tr>
<td>SQT</td>
<td>Quantity $\sqrt{zz' + \Omega^2}$ in modified Cheng-Kemp calculation; $\sqrt{zz'}$ in unmodified calculation</td>
</tr>
<tr>
<td>SQTPl</td>
<td>$\sqrt{\pi}$</td>
</tr>
<tr>
<td>SSA</td>
<td>$S \sin \alpha$</td>
</tr>
<tr>
<td>SSASQ</td>
<td>$(S \sin \alpha)^2$</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>TAU</td>
<td>Parameter $\tau = 0.664 + 1.72 \frac{h_w}{h_0}$</td>
</tr>
<tr>
<td>TAUN</td>
<td>$\tau N$</td>
</tr>
<tr>
<td>TAUN2</td>
<td>$(\tau N)^2$</td>
</tr>
<tr>
<td>TAUN3</td>
<td>$(\tau N)^3$</td>
</tr>
<tr>
<td>TAUN4</td>
<td>$(\tau N)^4$</td>
</tr>
<tr>
<td>TAUN6</td>
<td>$(\tau N)^6$</td>
</tr>
<tr>
<td>TINF</td>
<td>Free-stream temperature, $T_1$</td>
</tr>
<tr>
<td>TN</td>
<td>Diameter of leading edge, inch</td>
</tr>
<tr>
<td>TRATIO</td>
<td>Ratio of wedge surface temperature to free-stream temperature</td>
</tr>
<tr>
<td>TREF</td>
<td>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</td>
</tr>
<tr>
<td>UNIF</td>
<td>Free-stream velocity, $u_1$</td>
</tr>
<tr>
<td>UN(J)</td>
<td>Hollerith array for labelling the output as being based on the modified or unmodified Cheng-Kemp theory</td>
</tr>
<tr>
<td>VMUINF</td>
<td>Free-stream viscosity, lb/ft·sec</td>
</tr>
<tr>
<td>WK2</td>
<td>Square of leading-edge drag coefficient, $k^2$</td>
</tr>
<tr>
<td>WK3</td>
<td>$k^3$</td>
</tr>
<tr>
<td>WK4</td>
<td>$k^4$</td>
</tr>
<tr>
<td>WKVL</td>
<td>Next distance from the leading edge in the specified uniform sequence, inch</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<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 $IT$th specified distance from the leading edge, inch</td>
</tr>
<tr>
<td>YS(IT)</td>
<td>Shock ordinate at the $IT$th distance from the leading edge, inch</td>
</tr>
<tr>
<td>YSFAC</td>
<td>Coefficient $\gamma^2 k^3 / 8 A^4 \epsilon$ in calculation of shock ordinate</td>
</tr>
<tr>
<td>YSFAK</td>
<td>Coefficient $YSFAC \cdot t / (\tau N)^4$ 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 $16 A^7 \epsilon / \gamma 3k^4$</td>
</tr>
<tr>
<td>ZFAK</td>
<td>Coefficient $ZFAC \cdot (\tau N)^6 / t$</td>
</tr>
<tr>
<td>ZTA(IT)</td>
<td>$\zeta$ at the $IT$th 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>
zzps z..' as a single-precision value

4.135 subroutine weso\n
b0 coefficient $\sqrt[3]{6}$ 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 weso\n
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 $d \lambda / d \xi$ for newton-raphson iteration
Iteration counter

$N$

$P1$, $P2$, $P12$

$\zeta^{1/3}$, $\zeta^{1/2}$, $\zeta^{2/3}$

$Q1$, $Q2$, $Q3$, $Q4$, $Q5$

$1 + \sqrt{\lambda}$, $(1 + \sqrt{\lambda})^2$, $(1 + \sqrt{\lambda})^3$, $(1 + \sqrt{\lambda})^4$, $\ln (1 + \sqrt{\lambda})$

$SQTL$

$\sqrt{\lambda}$

$XL$

$\lambda$

$Z$

Nondimensional shock ordinate, $z$

$ZETA$

Nondimensional coordinate $\zeta$ parallel to the direction of free-stream flow

$ZT$

$\zeta$ calculated using trial value of $\lambda$

$ZZP$, $ZZPP$

$zz'$, $(zz')''$

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

LQ 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

II1  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

IQL  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
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
II

Absolute value of IZ for a species

IZ(L)

Charge of species L in units of electron charge (Note that IZ is negative for positively charged species)

J

Index indicating charge for species for which cross sections are to be supplied (see Table III)

JJ

Index indicating whether cross sections are to be supplied for electrons (see Table III)

KK

Index indicating charge of species for which cross sections are to be supplied (see Table III)

L

Step in cross section computations; species index

LL

Computational step at which additional cross section data are supplied for neutral-neutral interactions

NEWMV

Total number of parameters in V array

NN

Number of species for which cross section data are to be supplied
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 equivalent 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 equivalent to the
variable CX in unlabelled common. Subroutines GEΩMAR, ØUT1, and TRANS 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>Routines</th>
<th>Index</th>
<th>Routines</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>OUT1</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>AXSFCT</td>
<td>43</td>
<td>PUTQIN</td>
</tr>
<tr>
<td>9</td>
<td>AXSFCT</td>
<td>44</td>
<td>QCQUL</td>
</tr>
<tr>
<td>10</td>
<td>AXSFCT</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>CASSCF</td>
<td>47</td>
<td>QINTRP</td>
</tr>
<tr>
<td>13</td>
<td>CASSCF</td>
<td>48</td>
<td>QLJ</td>
</tr>
<tr>
<td>14</td>
<td>CASSCF</td>
<td>49</td>
<td>QMIX</td>
</tr>
<tr>
<td>15</td>
<td>DUMP</td>
<td>50</td>
<td>QREPP</td>
</tr>
<tr>
<td>16</td>
<td>DUMP</td>
<td>51</td>
<td>QSAME</td>
</tr>
<tr>
<td>17</td>
<td>DUMP</td>
<td>52</td>
<td>QTAQ</td>
</tr>
<tr>
<td>18</td>
<td>DUMP</td>
<td>53</td>
<td>Q11</td>
</tr>
<tr>
<td>19</td>
<td>DUMP</td>
<td>54</td>
<td>Q12</td>
</tr>
<tr>
<td>20</td>
<td>DUMP</td>
<td>55</td>
<td>Q13</td>
</tr>
<tr>
<td>21</td>
<td>DUMP</td>
<td>56</td>
<td>Q14</td>
</tr>
<tr>
<td>22</td>
<td>DUMP</td>
<td>57</td>
<td>RADIUS</td>
</tr>
<tr>
<td>23</td>
<td>DUMP</td>
<td>58</td>
<td>READ</td>
</tr>
<tr>
<td>24</td>
<td>DUMP</td>
<td>59</td>
<td>RESTMP</td>
</tr>
<tr>
<td>25</td>
<td>DUMP</td>
<td>60</td>
<td>RXT</td>
</tr>
<tr>
<td>26</td>
<td>DUMP</td>
<td>61</td>
<td>SHOCK</td>
</tr>
<tr>
<td>27</td>
<td>DUMP</td>
<td>62</td>
<td>SIMO</td>
</tr>
<tr>
<td>28</td>
<td>DUMP</td>
<td>63</td>
<td>STUNTS</td>
</tr>
<tr>
<td>29</td>
<td>DUMP</td>
<td>64</td>
<td>THERM</td>
</tr>
<tr>
<td>30</td>
<td>DUMP</td>
<td>65</td>
<td>THROAT</td>
</tr>
<tr>
<td>31</td>
<td>DUMP</td>
<td>66</td>
<td>TRANSP</td>
</tr>
<tr>
<td>32</td>
<td>DUMP</td>
<td>67</td>
<td>WEDGE</td>
</tr>
<tr>
<td>33</td>
<td>DUMP</td>
<td>68</td>
<td>WFSGLN</td>
</tr>
<tr>
<td>34</td>
<td>DUMP</td>
<td>69</td>
<td>XSFTC</td>
</tr>
<tr>
<td>35</td>
<td>DUMP</td>
<td>70</td>
<td>XSFCT</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>AFENTS</td>
<td></td>
<td>1, 6, 16, 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, 26, 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>AMPF</td>
<td></td>
<td>8, 16</td>
<td></td>
</tr>
<tr>
<td>ANGLE</td>
<td>ROWEDG</td>
<td>2, 30, 58, 67</td>
<td></td>
</tr>
<tr>
<td>ARM</td>
<td></td>
<td>16, 23, 35</td>
<td></td>
</tr>
<tr>
<td>ARHA</td>
<td></td>
<td>16, 27, 35</td>
<td></td>
</tr>
<tr>
<td>ARRH</td>
<td></td>
<td>16, 27, 35</td>
<td></td>
</tr>
<tr>
<td>ASTAR</td>
<td>TRANS4</td>
<td>50, 59</td>
<td></td>
</tr>
<tr>
<td>ATPJ</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>B</td>
<td>TRANS3</td>
<td>28, 29, 66</td>
<td></td>
</tr>
<tr>
<td>BETA</td>
<td></td>
<td>16, 36, 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>10, 35</td>
<td></td>
</tr>
<tr>
<td>BLINT</td>
<td>BL</td>
<td>8, 16, 35</td>
<td></td>
</tr>
<tr>
<td>BLNFA</td>
<td></td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>BLVAR</td>
<td></td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>BMPA</td>
<td></td>
<td>11, 16, 58</td>
<td></td>
</tr>
<tr>
<td>P</td>
<td>TRANS3</td>
<td>28</td>
<td></td>
</tr>
<tr>
<td>BSTAR</td>
<td>TRANS4</td>
<td>50, 58</td>
<td></td>
</tr>
<tr>
<td>BTA</td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>BZERO</td>
<td>READAT</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>CAR</td>
<td></td>
<td>26, 30, 59</td>
<td></td>
</tr>
<tr>
<td>CARX</td>
<td></td>
<td>26, 33, 36, 42</td>
<td></td>
</tr>
<tr>
<td>CARXTH</td>
<td></td>
<td>16, 36</td>
<td></td>
</tr>
<tr>
<td>CAS</td>
<td>SS</td>
<td>16, 35</td>
<td></td>
</tr>
<tr>
<td>CATFAC</td>
<td>MOD'AR</td>
<td>8, 23, 42</td>
<td></td>
</tr>
<tr>
<td>CCHI</td>
<td>RENF</td>
<td>2, 16, 30, 32, 58</td>
<td></td>
</tr>
<tr>
<td>CCI</td>
<td></td>
<td>2, 30, 35, 58</td>
<td></td>
</tr>
<tr>
<td>CCPJ</td>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>COIJ</td>
<td></td>
<td>1, 8, 11, 13, 16, 21, 38, 39, 63, 64, 66</td>
<td></td>
</tr>
<tr>
<td>CEACT</td>
<td></td>
<td>11, 16, 30, 58</td>
<td></td>
</tr>
<tr>
<td>TABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINE</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>CF</td>
<td>/CONVR/</td>
<td>1: 16, 22, 38, 66,</td>
<td></td>
</tr>
<tr>
<td>CG1</td>
<td></td>
<td>11: 16, 20, 26, 33,</td>
<td></td>
</tr>
<tr>
<td>CGMU</td>
<td></td>
<td>16, 26, 33,</td>
<td></td>
</tr>
<tr>
<td>CGMV</td>
<td>/COLDSP/</td>
<td>1: 8, 59,</td>
<td></td>
</tr>
<tr>
<td>CH</td>
<td></td>
<td>11: 11, 22, 23, 24,</td>
<td></td>
</tr>
<tr>
<td>CHA</td>
<td></td>
<td>1: 8, 11, 16, 21, 23, 26, 32, 33, 35, 38, 41, 59, 60,</td>
<td></td>
</tr>
<tr>
<td>CHANAM</td>
<td>/ROUT/</td>
<td>32, 59,</td>
<td></td>
</tr>
<tr>
<td>CHAP</td>
<td>/TNERK/</td>
<td>35, 60,</td>
<td></td>
</tr>
<tr>
<td>CHI1</td>
<td></td>
<td>11: 16, 39, 42,</td>
<td></td>
</tr>
<tr>
<td>CHII</td>
<td></td>
<td>16, 20, 33,</td>
<td></td>
</tr>
<tr>
<td>CLN91</td>
<td></td>
<td>11: 16,</td>
<td></td>
</tr>
<tr>
<td>CLN92</td>
<td>/TNDT/</td>
<td>13: 14,</td>
<td></td>
</tr>
<tr>
<td>CLN94</td>
<td>/TNDC/</td>
<td>11: 13,</td>
<td></td>
</tr>
<tr>
<td>CM</td>
<td></td>
<td>1: 8, 28, 39, 38, 39,</td>
<td></td>
</tr>
<tr>
<td>CMN</td>
<td></td>
<td>1: 8, 11, 16, 21, 23, 26, 32, 33, 39,</td>
<td></td>
</tr>
<tr>
<td>CMN</td>
<td>/TNDT/</td>
<td>11: 13,</td>
<td></td>
</tr>
<tr>
<td>CMN</td>
<td>/TNDC/</td>
<td>11: 16,</td>
<td></td>
</tr>
<tr>
<td>CMW</td>
<td></td>
<td>16, 26, 30, 58,</td>
<td></td>
</tr>
<tr>
<td>CNWLD</td>
<td>/INNE/</td>
<td>8: 16, 35,</td>
<td></td>
</tr>
<tr>
<td>CP</td>
<td>/CHAN/</td>
<td>58: 16,</td>
<td></td>
</tr>
<tr>
<td>CPW ALL</td>
<td>/DL/</td>
<td>1: 8, 16, 32, 39, 67,</td>
<td></td>
</tr>
<tr>
<td>CP1</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP2</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP3</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP4</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP5</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP6</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP7</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP8</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CP9</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CPA</td>
<td>/CHAN/</td>
<td>2:</td>
<td></td>
</tr>
<tr>
<td>CRP</td>
<td></td>
<td>11: 16, 26, 27, 32,</td>
<td></td>
</tr>
<tr>
<td>CRB</td>
<td></td>
<td>16: 23, 26, 33,</td>
<td></td>
</tr>
<tr>
<td>CSM</td>
<td></td>
<td>16: 35,</td>
<td></td>
</tr>
<tr>
<td>CTA</td>
<td></td>
<td>11: 16, 35,</td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td></td>
<td>1: 8, 28, 39, 38, 39,</td>
<td></td>
</tr>
<tr>
<td>CTAP</td>
<td></td>
<td>35: 36,</td>
<td></td>
</tr>
<tr>
<td>CTAP1</td>
<td>/READAT/</td>
<td>1: 8, 11, 16, 20, 21, 23, 26, 27, 32, 35, 38, 39, 65, 66,</td>
<td></td>
</tr>
<tr>
<td>CTAP2</td>
<td>/READAT/</td>
<td>58:</td>
<td></td>
</tr>
<tr>
<td>CTR</td>
<td></td>
<td>16: 35, 60,</td>
<td></td>
</tr>
<tr>
<td>CTC</td>
<td></td>
<td>16:</td>
<td></td>
</tr>
<tr>
<td>CTD</td>
<td>/TNDC/</td>
<td>11: 13,</td>
<td></td>
</tr>
<tr>
<td>CTF</td>
<td></td>
<td>11: 13,</td>
<td></td>
</tr>
<tr>
<td>CTF1</td>
<td>/TNERK/</td>
<td>16: 35,</td>
<td></td>
</tr>
<tr>
<td>CTFX</td>
<td>/TNERK/</td>
<td>35, 60,</td>
<td></td>
</tr>
<tr>
<td>CTFXX</td>
<td></td>
<td>3: 16, 17,</td>
<td></td>
</tr>
<tr>
<td>CTMXX</td>
<td>/MASSFL/</td>
<td>3: 16, 17,</td>
<td></td>
</tr>
<tr>
<td>CTP</td>
<td></td>
<td>16: 64,</td>
<td></td>
</tr>
<tr>
<td>CTP1</td>
<td></td>
<td>16: 27,</td>
<td></td>
</tr>
<tr>
<td>CTTP</td>
<td></td>
<td>16: 38,</td>
<td></td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>1: 6, 11, 13, 14, 23, 24, 35, 38, 60, 65,</td>
<td></td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>1: 6, 11, 13, 14, 23, 24, 35, 38, 60, 65,</td>
<td></td>
</tr>
<tr>
<td>CXMAX</td>
<td>/READAT/</td>
<td>16: 23, 30, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>CXMAX1</td>
<td>/READAT/</td>
<td>2: 58,</td>
<td></td>
</tr>
<tr>
<td>C2</td>
<td></td>
<td>1: 2, 16, 32, 38, 58,</td>
<td></td>
</tr>
<tr>
<td>DATAP</td>
<td>/STAPOUT/</td>
<td>1: 2, 16, 32, 38, 58,</td>
<td></td>
</tr>
<tr>
<td>V. ABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINE</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td>DATEST /</td>
<td>/</td>
<td>16, 27, 35,</td>
<td></td>
</tr>
<tr>
<td>DBTEST /</td>
<td>/</td>
<td>16, 27,</td>
<td></td>
</tr>
<tr>
<td>DCA /</td>
<td>/</td>
<td>35,</td>
<td></td>
</tr>
<tr>
<td>DCHA /</td>
<td>TNERK /</td>
<td>8, 21, 60,</td>
<td></td>
</tr>
<tr>
<td>DCHL /</td>
<td>RENE /</td>
<td>7, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>DCHPAT /</td>
<td>RENE /</td>
<td>2, 35, 58,</td>
<td></td>
</tr>
<tr>
<td>DCI /</td>
<td>/</td>
<td>35,</td>
<td></td>
</tr>
<tr>
<td>DDDL /</td>
<td>ALNE /</td>
<td>8, 16,</td>
<td></td>
</tr>
<tr>
<td>DDEL /</td>
<td>NEQ /</td>
<td>4, 8, 16, 35, 42,</td>
<td></td>
</tr>
<tr>
<td>DELH /</td>
<td>BL /</td>
<td>1, 4, 6, 8, 13, 16, 23, 35, 38, 42, 65,</td>
<td></td>
</tr>
<tr>
<td>DELBL /</td>
<td>BLNE /</td>
<td>8, 16,</td>
<td></td>
</tr>
<tr>
<td>DELTAX /</td>
<td></td>
<td>16, 35, 58, 60,</td>
<td></td>
</tr>
<tr>
<td>DELTXX /</td>
<td>READAT /</td>
<td>2, 58,</td>
<td></td>
</tr>
<tr>
<td>DELEI /</td>
<td>READAT /</td>
<td>2, 35, 36, 58,</td>
<td></td>
</tr>
<tr>
<td>DELTM /</td>
<td>READAT /</td>
<td>2, 58,</td>
<td></td>
</tr>
<tr>
<td>DGF /</td>
<td>/</td>
<td>16,</td>
<td></td>
</tr>
<tr>
<td>DH /</td>
<td>TRANS3 /</td>
<td>21, 39, 39, 60,</td>
<td></td>
</tr>
<tr>
<td>DIAM /</td>
<td>BL /</td>
<td>29, 66,</td>
<td></td>
</tr>
<tr>
<td>DLOGX /</td>
<td>/</td>
<td>38,</td>
<td></td>
</tr>
<tr>
<td>DLOGA /</td>
<td>/</td>
<td>11, 13, 14, 21, 24, 35, 42, 60, 65,</td>
<td></td>
</tr>
<tr>
<td>DLOGR /</td>
<td>/</td>
<td>11, 10, 21, 24, 35, 42, 60,</td>
<td></td>
</tr>
<tr>
<td>DMAX /</td>
<td>TNERK /</td>
<td>35, 60,</td>
<td></td>
</tr>
<tr>
<td>DTE /</td>
<td>/</td>
<td>8, 16, 21, 42, 60,</td>
<td></td>
</tr>
<tr>
<td>DUM /</td>
<td>TCNE /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>DUX /</td>
<td>RDWEDG /</td>
<td>2, 30, 58, 67,</td>
<td></td>
</tr>
<tr>
<td>DXMODP /</td>
<td>MODPAR /</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DXMP /</td>
<td>MODPAR /</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DXP /</td>
<td>SWITCH /</td>
<td>8,</td>
<td></td>
</tr>
<tr>
<td>DXPM /</td>
<td>SWITCH /</td>
<td>16, 23, 35,</td>
<td></td>
</tr>
<tr>
<td>EEP /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EEPHP /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP1 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP9 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP10 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP12 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP2 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP3 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP4 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP5 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP6 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP7 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP8 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP9 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP10 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP11 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP12 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP13 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP14 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP15 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP16 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP17 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP18 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP19 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP20 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP21 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP22 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP23 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP24 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP25 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP26 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP27 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP28 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP29 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP30 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP31 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP32 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP33 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP34 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP35 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP36 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP37 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP38 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP39 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP40 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP41 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP42 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP43 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP44 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP45 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP46 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP47 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP48 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP49 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP50 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP51 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP52 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP53 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP54 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP55 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP56 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP57 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP58 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP59 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP60 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP61 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP62 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP63 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP64 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP65 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP66 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP67 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP68 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP69 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP70 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP71 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP72 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP73 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP74 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP75 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP76 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP77 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP78 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP79 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>EFP80 /</td>
<td>EELFM /</td>
<td>58,</td>
<td></td>
</tr>
<tr>
<td>ELJ /</td>
<td>/</td>
<td>27, 30, 58, 64,</td>
<td></td>
</tr>
<tr>
<td>ELM /</td>
<td>/</td>
<td>30, 58,</td>
<td></td>
</tr>
<tr>
<td>ENT /</td>
<td>/</td>
<td>16, 23, 41,</td>
<td></td>
</tr>
<tr>
<td>EPN1 /</td>
<td>TONEQ /</td>
<td>16, 23, 41,</td>
<td></td>
</tr>
<tr>
<td>EPNP /</td>
<td>FLFM /</td>
<td>16, 23, 41,</td>
<td></td>
</tr>
<tr>
<td>EPSP /</td>
<td>STAG /</td>
<td>16, 23, 41,</td>
<td></td>
</tr>
<tr>
<td>EPSLON /</td>
<td>STAG /</td>
<td>16, 23, 32, 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>EP1</td>
<td>/ELEM</td>
<td>2, 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>EP10</td>
<td>/ELEM</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EP2</td>
<td>/ELEM</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EP3</td>
<td>/ELFM</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EP4</td>
<td>/ELFM</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>/ELFM</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EP8</td>
<td>/ELEM</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EP9</td>
<td>/ELEM</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>EP10</td>
<td>/ERROR</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FTAI</td>
<td>/ERROR</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>ETAJ</td>
<td>/ERROR</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FACMP</td>
<td>/NEWMP</td>
<td>16, 34, 58</td>
<td></td>
</tr>
<tr>
<td>FACNM</td>
<td>/ROUT</td>
<td>38, 58</td>
<td></td>
</tr>
<tr>
<td>FAILED</td>
<td>/ERROR</td>
<td>11, 13, 16, 25, 60</td>
<td></td>
</tr>
<tr>
<td>FLEWIS</td>
<td>/TRPROP</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FLOW</td>
<td>/ROUT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FLUX</td>
<td>/ROUT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>FSTAG</td>
<td>/MONPAR</td>
<td>1, 16, 23, 33, 35, 36, 41</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, 16, 8, 9, 11, 21, 23, 32, 33, 35, 36, 38, 39, 60</td>
<td></td>
</tr>
<tr>
<td>GJ6A</td>
<td>/OUTPUT</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>GJ6I</td>
<td>/OUTPUT</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>GJMF</td>
<td>/OUTPUT</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>GMN</td>
<td>/OUTPUT</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>GP2R</td>
<td>/MIXT</td>
<td>2</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>GSQ</td>
<td>/RENF</td>
<td>2, 35, 58</td>
<td></td>
</tr>
<tr>
<td>GTFST</td>
<td>/RENF</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>H</td>
<td>/</td>
<td>16, 38, 59</td>
<td></td>
</tr>
<tr>
<td>HDFUX</td>
<td>/</td>
<td>30, 32, 35, 38, 58, 63</td>
<td></td>
</tr>
<tr>
<td>H'</td>
<td>/</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>HPL</td>
<td>/</td>
<td>2, 8, 16, 38</td>
<td></td>
</tr>
<tr>
<td>HS</td>
<td>/</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>HSTAG</td>
<td>/</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>HTFST</td>
<td>/</td>
<td>2, 35, 69</td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>/</td>
<td>7, 16, 36, 63, 69</td>
<td></td>
</tr>
<tr>
<td>IAMDIP</td>
<td>/</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>IATOM</td>
<td>/</td>
<td>69</td>
<td></td>
</tr>
<tr>
<td>IC</td>
<td>/</td>
<td>16, 32, 33, 38, 58</td>
<td></td>
</tr>
</tbody>
</table>

ORIGIN PAGE B
OF POOR QUALITY
<table>
<thead>
<tr>
<th>TABLE</th>
<th>COMMON</th>
<th>REFERENCING ROUTIN%</th>
</tr>
</thead>
<tbody>
<tr>
<td>I183</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I184</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I185</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I186</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I187</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I188</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I189</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I190</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I191</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I192</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I193</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I194</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I195</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I196</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I197</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I198</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>I199</td>
<td>/TRANI /</td>
<td></td>
</tr>
<tr>
<td>IJC5</td>
<td>/OLDSP /</td>
<td></td>
</tr>
<tr>
<td>151</td>
<td>1, 8, 32, 58, 69.</td>
<td></td>
</tr>
<tr>
<td>154</td>
<td>1, 16, 27, 33, 38, 42.</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IMP</td>
<td>/TAPOUT /</td>
<td>32, 38, 58, 69.</td>
</tr>
<tr>
<td>INFO</td>
<td>/</td>
<td>8, 9, 11, 16, 21, 23, 24, 27, 35, 38, 42.</td>
</tr>
<tr>
<td>INFOV</td>
<td>/</td>
<td>16, 27, 30, 58, 64.</td>
</tr>
<tr>
<td>INFOV</td>
<td>/</td>
<td>2, 58, 69.</td>
</tr>
<tr>
<td>INT</td>
<td>/</td>
<td>2, 58, 69.</td>
</tr>
<tr>
<td>IP</td>
<td>/</td>
<td>16, 58.</td>
</tr>
<tr>
<td>IGA</td>
<td>/</td>
<td>11, 16, 30, 58, 69.</td>
</tr>
<tr>
<td>PI</td>
<td>/</td>
<td>16, 27, 35, 58, 69.</td>
</tr>
<tr>
<td>IPOINT</td>
<td>/</td>
<td>8, 16, 27, 35, 58, 69.</td>
</tr>
<tr>
<td>I0</td>
<td>/</td>
<td>7, 43, 49, 55, 56, 63, 69.</td>
</tr>
<tr>
<td>IR</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>IRUN</td>
<td>/</td>
<td>1, 16, 58.</td>
</tr>
<tr>
<td>IS</td>
<td>/</td>
<td>27, 58, 69.</td>
</tr>
<tr>
<td>ISATOM</td>
<td>/</td>
<td>27, 58, 69.</td>
</tr>
<tr>
<td>ISC</td>
<td>/</td>
<td>16, 20, 21, 26, 30, 32, 33, 35, 36, 39, 58.</td>
</tr>
<tr>
<td>ISCTI</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISCPI</td>
<td>/</td>
<td>16, 21, 35, 39, 69.</td>
</tr>
<tr>
<td>ISEQ</td>
<td>/</td>
<td>3, 58, 63, 69.</td>
</tr>
<tr>
<td>ISAPF</td>
<td>/</td>
<td>16, 25, 58, 69.</td>
</tr>
<tr>
<td>ISMC</td>
<td>/</td>
<td>1, 16, 27, 29, 58, 69.</td>
</tr>
<tr>
<td>ISMCR</td>
<td>/</td>
<td>1, 16, 27, 33, 58, 69.</td>
</tr>
<tr>
<td>ISMOL</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISMOL</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISR</td>
<td>/</td>
<td>11, 16, 21, 30, 35, 38, 39, 58.</td>
</tr>
<tr>
<td>ISR1</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISS</td>
<td>/</td>
<td>1, 8, 12, 16, 20, 21, 23, 24, 27, 30, 32.</td>
</tr>
<tr>
<td>ISS1</td>
<td>/</td>
<td>33, 35, 38, 39, 41, 58, 60, 63, 64, 66, 69.</td>
</tr>
<tr>
<td>ISS2</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISS2R</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISS2R</td>
<td>/</td>
<td>58.</td>
</tr>
<tr>
<td>ISS2P1</td>
<td>/</td>
<td>16, 21, 35, 39, 60.</td>
</tr>
<tr>
<td>ISS2P2</td>
<td>/</td>
<td>16, 21, 35, 39, 69.</td>
</tr>
<tr>
<td>IABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td>ISSP3</td>
<td></td>
<td>16, 21, 35, 39, 60,</td>
</tr>
<tr>
<td>ISSP4</td>
<td></td>
<td>16, 21, 35, 60,</td>
</tr>
<tr>
<td>ISSPR</td>
<td></td>
<td>35,</td>
</tr>
<tr>
<td>ISW1A</td>
<td></td>
<td>1, 16, 58,</td>
</tr>
<tr>
<td>ISW1B</td>
<td></td>
<td>1, 16, 58, 63, 66,</td>
</tr>
<tr>
<td>ISW1A</td>
<td></td>
<td>1, 16, 58,</td>
</tr>
<tr>
<td>ISW1B</td>
<td></td>
<td>1, 16, 38, 59, 59,</td>
</tr>
<tr>
<td>ISW1A</td>
<td></td>
<td>1, 16, 58,</td>
</tr>
<tr>
<td>ISW1B</td>
<td></td>
<td>1, 6, 9, 13, 16, 23, 24, 30, 35, 38, 42, 58, 65,</td>
</tr>
<tr>
<td>ISW2A</td>
<td></td>
<td>1, 16, 58,</td>
</tr>
<tr>
<td>ISW2B</td>
<td></td>
<td>1, 8, 16, 58,</td>
</tr>
<tr>
<td>ISW3A</td>
<td></td>
<td>1, 16, 58, 59,</td>
</tr>
<tr>
<td>ISW3B</td>
<td></td>
<td>1, 11, 16, 35, 42, 59, 60,</td>
</tr>
<tr>
<td>ISW3A</td>
<td></td>
<td>1, 16, 58,</td>
</tr>
<tr>
<td>ISW3B</td>
<td></td>
<td>1, 16, 38, 58,</td>
</tr>
<tr>
<td>ISW7A</td>
<td>/READAT/</td>
<td>2, 58,</td>
</tr>
<tr>
<td>ISW7B</td>
<td>/ROTR/</td>
<td>3, 43, 47, 58, 66,</td>
</tr>
<tr>
<td>ISW9B</td>
<td>/RDWEDG/</td>
<td>2, 58, 67,</td>
</tr>
<tr>
<td>ITB</td>
<td></td>
<td>16, 58, 35,</td>
</tr>
<tr>
<td>ITPUT</td>
<td>/TAPOUT/</td>
<td>1, 2, 16, 32, 38, 58,</td>
</tr>
<tr>
<td>ITR</td>
<td>/TNCNEQ/</td>
<td>16, 19, 30, 58,</td>
</tr>
<tr>
<td>ITYPER</td>
<td>/TAPOUT/</td>
<td>1, 32, 78,</td>
</tr>
<tr>
<td>IUPOD</td>
<td></td>
<td>6, 16, 21, 35, 42, 58, 65,</td>
</tr>
<tr>
<td>IZERO</td>
<td></td>
<td>1, 16, 39, 58, 64,</td>
</tr>
<tr>
<td>1355</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>110</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>1100</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>111</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>112</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>113</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>114</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>115</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>116</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>117</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>118</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>119</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>12</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>120</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>121</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>122</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>123</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>124</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>125</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>126</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>127</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>128</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>129</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>13</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>130</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>131</td>
<td>/TRANI/</td>
<td>58,</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td>J0</td>
<td>/TRANJ</td>
<td>3, 58, 7, 43, 49, 55, 56, 63, 69, 359</td>
</tr>
<tr>
<td>J1</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J10</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J11</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J12</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J13</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J14</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J15</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J16</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J17</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J18</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J19</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J20</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J20</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J21</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J22</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J23</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J24</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>J25</td>
<td>/TRANJ</td>
<td>58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58, 58</td>
</tr>
<tr>
<td>TABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td>J71</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J72</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J73</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J74</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J75</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J76</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J77</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J78</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J79</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J80</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J81</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J82</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J83</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J84</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J85</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J86</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J87</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J88</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J89</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J90</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J91</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J92</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J93</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J94</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J95</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J96</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J97</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J98</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>J99</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K0</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K1</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K2</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K3</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K4</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K5</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K6</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K7</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K8</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>K9</td>
<td>TRANJ</td>
<td>58,</td>
</tr>
<tr>
<td>KDIM</td>
<td>MODPAR</td>
<td>2, 16, 30, 32, 58,</td>
</tr>
<tr>
<td>KKQ</td>
<td>TRANS2</td>
<td>3, 58, 63, 69,</td>
</tr>
<tr>
<td>KQ</td>
<td>TRANS7</td>
<td>10, 43, 63, 69,</td>
</tr>
<tr>
<td>KTE</td>
<td>TNJNF0</td>
<td>11, 16, 30, 59,</td>
</tr>
<tr>
<td>KTR</td>
<td>TNOF0</td>
<td>11, 16, 30, 59,</td>
</tr>
<tr>
<td>KUR</td>
<td></td>
<td>11, 30, 58,</td>
</tr>
<tr>
<td>LC</td>
<td></td>
<td>16, 35,</td>
</tr>
<tr>
<td>LEWIS</td>
<td>ROMOD</td>
<td>2, 30, 34, 58,</td>
</tr>
<tr>
<td>LIMOUT</td>
<td>ROMOUT</td>
<td>38, 58,</td>
</tr>
<tr>
<td>LDIJ</td>
<td></td>
<td>21, 26, 30, 32, 35, 39, 58, 69,</td>
</tr>
<tr>
<td>LSS</td>
<td>TRANS5</td>
<td>69,</td>
</tr>
<tr>
<td>MBL</td>
<td>MCPT</td>
<td>24, 58,</td>
</tr>
<tr>
<td>MHP</td>
<td>ROMAIN</td>
<td>1, 58,</td>
</tr>
<tr>
<td>MDLPT</td>
<td>MODP2</td>
<td>8, 35, 42,</td>
</tr>
<tr>
<td>M1</td>
<td>TRANS8</td>
<td>16, 20, 33, 36,</td>
</tr>
<tr>
<td>N</td>
<td>TRANS8</td>
<td>43, 69,</td>
</tr>
<tr>
<td>NSAGE</td>
<td>ROWDAG</td>
<td>2, 30, 58, 67,</td>
</tr>
<tr>
<td>NPL</td>
<td>AREA</td>
<td>8, 16, 38, 58,</td>
</tr>
<tr>
<td>NCS</td>
<td>COLDSP</td>
<td>1, 8, 32, 59,</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td>PRESSAI</td>
<td>READAT</td>
<td>50,</td>
</tr>
<tr>
<td>PRESSB</td>
<td></td>
<td>16, 23,</td>
</tr>
<tr>
<td>PESTH</td>
<td></td>
<td>1, 16, 23, 36,</td>
</tr>
<tr>
<td>PRF</td>
<td>TRPROP</td>
<td>60,</td>
</tr>
<tr>
<td>PRD</td>
<td>RESPRP</td>
<td>9, 16, 35, 39, 42,</td>
</tr>
<tr>
<td>PRREF</td>
<td>BLOUT</td>
<td>1,</td>
</tr>
<tr>
<td>QT</td>
<td>TRANS1</td>
<td>2, 5, 16, 38,</td>
</tr>
<tr>
<td>QTFC</td>
<td>TNCF</td>
<td>11, 16, 21, 34,</td>
</tr>
<tr>
<td>QTPR</td>
<td>TNCF</td>
<td>11, 16, 21, 34,</td>
</tr>
<tr>
<td>QEVAX</td>
<td>TNERK</td>
<td>16,</td>
</tr>
<tr>
<td>QW</td>
<td></td>
<td>16, 20, 26, 33,</td>
</tr>
<tr>
<td>QPJ</td>
<td>COLDSP</td>
<td>1, 8, 12, 58,</td>
</tr>
<tr>
<td>QP</td>
<td></td>
<td>11, 16, 39, 58,</td>
</tr>
<tr>
<td>QTFST</td>
<td>RENE</td>
<td>2, 35, 58,</td>
</tr>
<tr>
<td>QDOTT</td>
<td>BLOUT</td>
<td>2, 8, 16, 38,</td>
</tr>
<tr>
<td>QACLE</td>
<td>ROWEEDG</td>
<td>2, 30, 55, 67,</td>
</tr>
<tr>
<td>QADG</td>
<td>READAT</td>
<td>2, 55,</td>
</tr>
<tr>
<td>READXS</td>
<td>READAT</td>
<td>2, 58,</td>
</tr>
<tr>
<td>REPQ</td>
<td>BLOUT</td>
<td>2, 16, 38,</td>
</tr>
<tr>
<td>RH</td>
<td></td>
<td>8,</td>
</tr>
<tr>
<td>RHP</td>
<td></td>
<td>1, 8, 11, 16, 21, 26, 32, 38, 39, 59,</td>
</tr>
<tr>
<td>RHO</td>
<td></td>
<td>1, 9, 11, 16, 21, 23, 24, 32, 33, 35, 36, 38, 39,</td>
</tr>
<tr>
<td>ROA</td>
<td></td>
<td>16, 23, 35,</td>
</tr>
<tr>
<td>RCOBAR</td>
<td></td>
<td>16, 23, 33, 41,</td>
</tr>
<tr>
<td>RHOINF</td>
<td></td>
<td>16,</td>
</tr>
<tr>
<td>RHOINF</td>
<td>OUTPUT</td>
<td>67,</td>
</tr>
<tr>
<td>RHP</td>
<td></td>
<td>11, 16, 21,</td>
</tr>
<tr>
<td>RHCN</td>
<td></td>
<td>11, 16,</td>
</tr>
<tr>
<td>RHTH</td>
<td></td>
<td>1, 16, 23, 24, 35,</td>
</tr>
<tr>
<td>RCPARA</td>
<td></td>
<td>11, 16, 23, 26, 33, 4,</td>
</tr>
<tr>
<td>RCDRP</td>
<td></td>
<td>16, 26, 33, 41,</td>
</tr>
<tr>
<td>RORP</td>
<td>REAC</td>
<td>*58,</td>
</tr>
<tr>
<td>RP1</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP11</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP12</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP13</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP14</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP15</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP16</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP17</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP18</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP19</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP20</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP21</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP22</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP23</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>RP24</td>
<td>REAC</td>
<td>2, 58,</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td>SKI1</td>
<td></td>
<td>16, 20, 33</td>
</tr>
<tr>
<td>SL</td>
<td></td>
<td>16, 35, 58</td>
</tr>
<tr>
<td>SL64</td>
<td></td>
<td>16, 35, 38</td>
</tr>
<tr>
<td>SM</td>
<td></td>
<td>16, 23, 24, 35, 36, 38, 59</td>
</tr>
<tr>
<td>SMASS</td>
<td>MASSFL</td>
<td>2, 8, 16, 38</td>
</tr>
<tr>
<td>SU</td>
<td>ALOUT</td>
<td>27, 58, 63</td>
</tr>
<tr>
<td>S1</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S10</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S11</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S12</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S13</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S14</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S15</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S16</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S17</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S18</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S19</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S20</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S21</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S22</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S23</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S24</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S25</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S26</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S27</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S28</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S29</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S30</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S31</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S32</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S33</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S34</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S35</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S36</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S37</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S38</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S39</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S40</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S41</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S42</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S43</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S44</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S45</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S46</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S47</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S48</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S49</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S50</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S51</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S52</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S53</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S54</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
<tr>
<td>S55</td>
<td>SPC</td>
<td>2, 58</td>
</tr>
</tbody>
</table>

**ORIGINAL PAGE 13**

**OF POOR QUALITY**
<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>COMMON</th>
<th>REFERENCING ROUTINES</th>
</tr>
</thead>
<tbody>
<tr>
<td>T97</td>
<td>OUTMOD</td>
<td>32, 38</td>
</tr>
<tr>
<td>T2ST</td>
<td></td>
<td>16, 20, 27, 33</td>
</tr>
<tr>
<td>T2STAB</td>
<td></td>
<td>16, 23, 27, 36</td>
</tr>
<tr>
<td>T2ST2</td>
<td>RENE</td>
<td>2, 35, 59</td>
</tr>
<tr>
<td>T2A</td>
<td></td>
<td>16, 30, 58, 64</td>
</tr>
<tr>
<td>T2B</td>
<td></td>
<td>16, 27, 30, 58, 64</td>
</tr>
<tr>
<td>T2C</td>
<td></td>
<td>16, 27, 30, 58, 64</td>
</tr>
<tr>
<td>T2D</td>
<td></td>
<td>16, 27, 30, 58, 64</td>
</tr>
<tr>
<td>T2E</td>
<td></td>
<td>16, 30, 58, 64</td>
</tr>
<tr>
<td>T2F</td>
<td></td>
<td>2, 8, 10, 38</td>
</tr>
<tr>
<td>T2G</td>
<td></td>
<td>16, 27, 30, 58, 64</td>
</tr>
<tr>
<td>T2H</td>
<td>POLYAT</td>
<td>27, 30, 59, 64</td>
</tr>
<tr>
<td>T2HVP</td>
<td>OUTPUT</td>
<td>30, 58</td>
</tr>
<tr>
<td>T2IF</td>
<td>OUTPUT</td>
<td>67</td>
</tr>
<tr>
<td>TL</td>
<td>TRANS4</td>
<td>47, 58</td>
</tr>
<tr>
<td>TL1</td>
<td>TRANS4</td>
<td>40, 59</td>
</tr>
<tr>
<td>TL10</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL11</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL12</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL13</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL14</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL15</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL16</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL17</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL18</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL19</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL2</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL20</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL21</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL22</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL23</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL24</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL25</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL26</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL27</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL28</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL29</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL3</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL30</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL31</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL32</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL33</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL34</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL35</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL36</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL37</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL38</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>TL39</td>
<td>TRANS4</td>
<td>3</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td>----------------------</td>
</tr>
<tr>
<td>TL4</td>
<td>/TRANS4/</td>
<td>3, 16, 32, 58</td>
</tr>
<tr>
<td>TL40</td>
<td>/TRANS4/</td>
<td>3</td>
</tr>
<tr>
<td>TL5</td>
<td>/TRANS4/</td>
<td>3</td>
</tr>
<tr>
<td>TL6</td>
<td>/TRANS4/</td>
<td>3</td>
</tr>
<tr>
<td>TL7</td>
<td>/TRANS4/</td>
<td>3</td>
</tr>
<tr>
<td>TL8</td>
<td>/TRANS4/</td>
<td>3</td>
</tr>
<tr>
<td>TL9</td>
<td>/TRANS4/</td>
<td>3</td>
</tr>
<tr>
<td>TMODEL</td>
<td>/MODPAR/</td>
<td>2, 16, 32, 32, 58</td>
</tr>
<tr>
<td>TNFP</td>
<td>/TNEF/</td>
<td>58</td>
</tr>
<tr>
<td>TN1</td>
<td>/TNEF/</td>
<td>2</td>
</tr>
<tr>
<td>TN2</td>
<td>/TNEF/</td>
<td>2</td>
</tr>
<tr>
<td>TPLATEF</td>
<td>/MODPAR/</td>
<td>2, 30, 58</td>
</tr>
<tr>
<td>TPRINT</td>
<td>/READAT/</td>
<td>2, 58</td>
</tr>
<tr>
<td>TPROP</td>
<td>/TPROPP/*</td>
<td>63</td>
</tr>
<tr>
<td>TSTOP</td>
<td>/TPROPP/*</td>
<td>8</td>
</tr>
<tr>
<td>TS</td>
<td>/STAG/</td>
<td>16, 32, 58</td>
</tr>
<tr>
<td>T5AR</td>
<td>/MODRT/</td>
<td>16, 34, 39, 58</td>
</tr>
<tr>
<td>TSDIAM</td>
<td>/MODRT/</td>
<td>2, 16, 39, 58</td>
</tr>
<tr>
<td>T5F</td>
<td>/STAG/</td>
<td>16</td>
</tr>
<tr>
<td>TSTOP</td>
<td>/STAG/</td>
<td>16, 23, 27, 35, 58</td>
</tr>
<tr>
<td>TSTOP1</td>
<td>/MASSFL/</td>
<td>2, 27, 58</td>
</tr>
<tr>
<td>TT</td>
<td>/TENE/</td>
<td>2, 35, 58</td>
</tr>
<tr>
<td>TTOPST</td>
<td>/TENE/</td>
<td>2, 35, 58</td>
</tr>
<tr>
<td>TWALL</td>
<td>/AL/</td>
<td>1, 2, 3, 16, 39, 58</td>
</tr>
<tr>
<td>TWEDGE</td>
<td>/MEDG/</td>
<td>2, 30, 54, 67</td>
</tr>
<tr>
<td>T22</td>
<td>/STAG/</td>
<td>23, 32, 35</td>
</tr>
<tr>
<td>T2F</td>
<td>/STAG/</td>
<td>23, 32, 35</td>
</tr>
<tr>
<td>U</td>
<td>/OUT/</td>
<td>67</td>
</tr>
<tr>
<td>UINF</td>
<td>/OUT/</td>
<td>67</td>
</tr>
<tr>
<td>UP</td>
<td>/OUT/</td>
<td>16, 35, 42</td>
</tr>
<tr>
<td>US</td>
<td>/SS/</td>
<td>1, 8, 21, 23, 42</td>
</tr>
<tr>
<td>V</td>
<td>/TRANS7/</td>
<td>10, 43, 63, 69</td>
</tr>
<tr>
<td>V1SC</td>
<td>/TPROPP/</td>
<td>1, 6, 16, 23, 32, 39, 66, 67</td>
</tr>
<tr>
<td>V1SCR</td>
<td>/TPROPP/</td>
<td>1, 6, 16, 23, 32, 39, 66, 67</td>
</tr>
<tr>
<td>V1ROT</td>
<td>/AL/</td>
<td>1, 16</td>
</tr>
<tr>
<td>VMUINF</td>
<td>/OUTPUT/</td>
<td>67</td>
</tr>
<tr>
<td>VV</td>
<td>/TRAV/</td>
<td>63, 69</td>
</tr>
<tr>
<td>VV1</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV10</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV11</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV12</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV13</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV14</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV15</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV16</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV17</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV18</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VV19</td>
<td>/TRAV/</td>
<td>3</td>
</tr>
<tr>
<td>VARIABLE</td>
<td>COMMON</td>
<td>REFERENCING ROUTINES</td>
</tr>
<tr>
<td>-----------</td>
<td>----------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>XNU1</td>
<td>/</td>
<td>11, 16, 35, 58,</td>
</tr>
<tr>
<td>XNU1 J</td>
<td>/</td>
<td>11, 30, 35, 58,</td>
</tr>
<tr>
<td>XNU1 JP</td>
<td>/</td>
<td>11, 30, 35, 58,</td>
</tr>
<tr>
<td>XP</td>
<td>/SWITCH/</td>
<td>8, 23,</td>
</tr>
<tr>
<td>XPR</td>
<td>/SWITCH/</td>
<td>16, 23, 35,</td>
</tr>
<tr>
<td>XSN</td>
<td>/BOUT/</td>
<td>2, 8, 16, 35, 39,</td>
</tr>
<tr>
<td>XXX</td>
<td>/TAOUT/</td>
<td>1, 38,</td>
</tr>
<tr>
<td>XZER01</td>
<td>/BL/</td>
<td>8, 16, 23, 58,</td>
</tr>
<tr>
<td>XZER01</td>
<td>/READA/</td>
<td>58,</td>
</tr>
<tr>
<td>YOZO</td>
<td>/BLRAD/</td>
<td>8, 57,</td>
</tr>
<tr>
<td>ZCAP</td>
<td>/EQC/</td>
<td>16, 20, 26, 32,</td>
</tr>
<tr>
<td>ZCH</td>
<td>/EQC/</td>
<td>16, 20, 26, 32,</td>
</tr>
<tr>
<td>ZCH</td>
<td>/EQC/</td>
<td>16, 20, 26, 32,</td>
</tr>
<tr>
<td>ZCMU</td>
<td>/EOC2/</td>
<td>2, 26,</td>
</tr>
<tr>
<td>ZMP2</td>
<td>/TRANSI/</td>
<td>28, 65, 69,</td>
</tr>
<tr>
<td>ZP</td>
<td>/</td>
<td>16, 33, 41,</td>
</tr>
<tr>
<td>ZPA</td>
<td>/</td>
<td>16, 23, 26, 33,</td>
</tr>
<tr>
<td>ZP80</td>
<td>/NOZZ/</td>
<td>58,</td>
</tr>
<tr>
<td>ZP2Z</td>
<td>/EOC2/</td>
<td>22, 26,</td>
</tr>
<tr>
<td>ZP19</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP10</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP11</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP12</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP13</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP14</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP15</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP16</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP17</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP18</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP19</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP2</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP30</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP3</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP4</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP5</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP6</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP7</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP8</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>ZP9</td>
<td>/NOZZ/</td>
<td>2,</td>
</tr>
<tr>
<td>Z800</td>
<td>/EQC/</td>
<td>16, 20, 26,</td>
</tr>
<tr>
<td>ZRH0</td>
<td>/EQC/</td>
<td>16, 20, 26, 32,</td>
</tr>
<tr>
<td>ZSN</td>
<td>/EQC/</td>
<td>16, 20, 26, 32,</td>
</tr>
</tbody>
</table>
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.
COMMON /OUTPUT/ FOUT(35),GFNF(20)
COMMON /RDOUT/ FLOW,FACNAM,CHANAM,LIMOL
COMMON /SS/ CAS,US
COMMON /SPEC/ SPRP(43,30)
COMMON /LN/ ISATOM,ISMOL,JATOM,JMOL
COMMON /EELEM/ EEP(2,10)
COMMON /THRT/ RSA
DATA NCASE,NERRDR,NRCTOT,IRCO /4*0/, ONE /1*1/, PIV /3*1A159/
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)
ISW3D=IS3
CALL READ
IS3-ISW3D
IF (.NOT.*NOTRAN) GO TO 20
ISW3D=0
VISC=0.
PR=0.
SIGMA=0.
FLEWIS=0.
20 CALL LIST
IF (.NOT.*NOTRAN) WRITE (6,330)
ET=ETO
CALL ELTIME (ETO,1)
NRDOUT=0
NCASE=NCASE+1
ERR=FALSE.
OMDSTT1)=1.
OMDSTT2)=1.
IF (ISW28*LE0) SM=0.
CALL INGAS
IF (ERR) GO TO 290
IF (ISW28*LE0) CALL RESTOP
IF (ERR) GO TO 290
IF (ISW28*LT0) GO TO 30
CALL INIT
CALL INTA
IF (ERR) GO TO 290
IF (ISW6A*GE0) GO TO 40
CALL STUNTS
IRCO=1
GO TO 300
RHO = \rho_{\text{le}}

\text{CH} = \text{CHA}

\text{CM} = \text{CMA}

\text{WRITE} (5.3703)

\text{CALL OUT1}

IF \text{(ERR)} \text{GO TO} 290

\text{CALL ELTIME} (ET.1)

IF (\text{ISW6A.EQ.2} \text{OR} \text{ISW6B.EQ.0}) \text{GO TO} 300

IF (\text{NOTPAN}) \text{GO TO} 160

\text{CALL TRANSP} (\text{CTAPE})

IF (\text{ERR}) \text{GO TO} 290

\text{VISCC} = \text{VISC} \times \text{CF}(3)

\text{WRITE} (6+380)

\text{CALL OUT1}

\text{CALL ELTIME} (ET.1)

IF (\text{ISW6A.NE.0}) \text{GO TO} 300

\text{VISCR} = \text{VISC}

\text{PRH} = \text{PR}

\text{SIGH} = \text{SIGMA}

\text{FLEW} = \text{FLEW}

\text{IS} = \text{VISROT} = \text{VIS} / \text{CTAP}

\text{HOP} = \text{CHA} \times \text{CRP} / \text{CMA}

\text{Sk} = \text{CPbALL} \times \text{TWALL} / \text{HOP} - 1.

IF (\text{ISW1B.EQ.1}) \text{GO TO} 190

IF (\text{NOT DATAPE}) \text{GO TO} 180

\text{ISOLN} = 1

\text{ITYPER} = 1

\text{XXX}(1) = \text{NOZZLE} + 0.1

\text{XXX}(2) = \text{FLOW}

\text{XXX}(3) = \text{PHESA}

\text{XXX}(4) = \text{CF} \times (\text{SI} \times \text{CHA})

\text{XXX}(5) = \text{IAM} (1)

\text{XXX}(4) = \text{<TAP}

\text{XXX}(7) = \text{CF} (1)

\text{XXX}(9) = \text{SEN}

\text{XXX}(9) = \text{CMA}

\text{XXX}(10) = \text{DIAM} (2)

\text{NRCOUT} = \text{NWCOUT} + 1

\text{WASTE} (\text{ITPOUT})

\text{ITYPER = ISOLN} (\text{XXX}(1) \text{< 0.1})

\text{GO TO} (180 \text{* 210} \text{* 280})

\text{CALL FROZEN}

IF (\text{EDR}) \text{GO TO} 290

\text{CT} = \text{CfMAX}

\text{PRES} = \text{PRESTH}

\text{RHO} = \text{WHTH}

\text{SU} = \text{SM/RHTH}

\text{CH} = \text{CHA} - 0.5 \times \text{SU}^2

\text{WRITE} (4.3901)

\text{CALL OUT1}

\text{CALL ELTIME} (ET.1)

\text{CALL NRMAX}

\text{IF (ERR)} \text{GO TO} 290

\text{PRESTH} = \text{PRES}

\text{UTH} = \text{SU}

\text{CHTH} = \text{CH}

\text{CMTH} = \text{CMTH}
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
IM=1
JK=0
isman=ismc
ISNR=ISS
WRITE (6,400)
CT=CTMAX
PRES=PRETH
RHO=RHTH
SU=UTH
CH=CHTH
CM=CMTH
DO 230 I=1,ISS
SAVEC(I)=GMF(I)
CALL OUT1
IF (ERR) GO TO 290
CALL ELTIME (ET*1)
IF (ISW2AO.EQ.ZERO) GO TO 300
DO 240 I=1,ISS
GJ(I)=GJA(I)
WRITE (6,400)
PRES=PRETH
N=0
SUM1=ALOG(RHTH)
A=4*SUM1
N=i+1
IF (N-50) 260,270,270
SUM2=A+2
SUM3=RHTH**2
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**2)
DELT=DELT1
IF (ISW3B .NE. 0) DELT=0.049*(1-CTMAX)
CT=i+0-DELT
CM=CMTH
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 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
ELMIN=(ET-ETO)/60.
WRITE (6,410) ELMIN
IF (ISW4A) 10,310,10
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)
FORMAT (20H DENSITY FIT-ALPHA=*,PE15.7,10H CONSTANT=*,E15.7)
FORMAT (15.16H CASES COMPLETED/15.13H CASES FAILED)
FORMAT (27H SPECIFIC HEAT OF COLD GAS=*,F7.4,16H BTU/LB-DEG R AT, F)
FORMAT (1.2+5.6 DEG K)
FORMAT (1H1.9X,24H- RESERVOIR CONDITIONS -)
FORMAT (1H1.9X,24H- EQUILIBRIUM THROAT CONDITIONS -)
FORMAT (1H1.7X,28H- FROZEN THROAT CONDITIONS -)
FORMAT (1H1.6X,33H- EQUILIBRIUM THROAT CONDITIONS -)
FORMAT (24H CASE EXECUTION TIME WAS*,F6.2,6H MINUTES)
FORMAT (8HORUN NO.,17.5X,4HCASE,1A,23H OF THIS JOB COMPLETED*,15.2)
14H RECORDS WRITTEN ON TAPE,13,1H,,17,36H RECORDS WRITTEN SO FAR IN
2 THIS JOB*)
END
BLOCK DATA
LOGICAL DATAE, WEDGEM, AXISYM, READXS, RE, AAMS, AXIMOD
DOUBLE PRECISION AIN
EP8(2), EP9(2), EP10(2)
COMMON /SPEC/ SP1(43), SP2(43), SP3(43), SP4(43), SP5(43), SP6(43),
SP7(43), SP8(43), SP9(43), SP10(43), SP11(43), SP12(43), SP13(43),
SP14(43), SP15(43), SP16(43), SP17(43), SP18(43), SP19(43),
SP20(43), SP21(43), SP22(43), SP23(43), SP24(43), SP25(43),
SP26(43), SP27(43), SP28(43), SP29(43), SP30(43),
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),
RP14(29), RP15(29), RP16(29), RP17(29), RP18(29), RP19(29),
RP20(29), RP21(29), RP22(29), RP23(29), RP24(29), RP25(29),
RP26(29), RP27(29), RP28(29), RP29(29), RP30(29), RP31(29),
RP32(29), RP33(29), RP34(29), RP35(29), RP36(29), RP37(29),
RP38(29), RP39(29), RP40(29), RP41(29), RP42(29), RP43(29),
RP44(29), RP45(29), RP46(29), RP47(29), RP48(29), RP49(29),
RP50(29), RP51(29), RP52(29), RP53(29), RP54(29), RP55(29),
RP56(29), RP57(29), RP58(29), RP59(29), RP60(29), RP61(29),
RP62(29), RP63(29), RP64(29), RP65(29), RP66(29), RP67(29),
RP68(29), RP69(29), RP70(29), RP71(29), RP72(29), RP73(29),
RP74(29), RP75(29), RP76(29), RP77(29), RP78(29), RP79(29),
RP80(29), RP81(29), RP82(29), RP83(29), RP84(29), RP85(29),
RP86(29), RP87(29), RP88(29), RP89(29), RP90(29), RP91(29),
RP92(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),
NSECU(2), ISHAPE(12,2), NRCLLS, NBL
COMMON /MASSF/ SMAS, CTMXX, TSTP, JS(20)
COMMON /BL/ DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
SWRO, JDIM, IPCINT
COMMON /MOPAR/ XMPI, DXM, FSTAG, CATFAC, TMODEL, XMODP1, DXXMODP,
TPLATE, KDI
COMMON /RLIST/ IGAS, IGASO, NOZZO, ICHAN
COMMON /MOPDT/ TS2MI20, TSAR(20), NTB, MBL
COMMON /TAPOUT/ XXX(10), IPOUT, NRCCOUT, IFLOW, ITYPE, IMP, DATAE
COMMON /BLOUT/ REFP, THETA(2), SN(2), XSN(2), PPREF, HR, QWDOT(2),
TAUW(2)
COMMON /NEWMP/ FACMP, NMMPD
COMMON /RDWEDG/ ANGLE(10), RADLE(5), WX1, DXW, WX(20), TWEDGE, WK,
NWX, NANGLE, NRADLE, WEDGEM, AXISYM, ISWHB
COMMON /CHAN/ CP1(5), CP2(5), CP3(5), CP4(5), CP(5),
COMMON /AVG/ WSAYE
COMMON /TNE/ TN1(186), TN2(186)
COMMON /RDMDP/ LEWIS, IAMPI
COMMON /READAT/ PRXAI, ETLL1, TPRNT, DELTIX, CMAXI,
COMMON /HSTAG/ READS, READA, AAMS, AXIMOD
COMMON /ICASE/ NEELS, ISWB7, INT, ICHAN, NQSI, INEQVI, NREC, JCS(10).
ISCI, ISSI, ISRI, ICI, IE(10), IR(6)

COMMON /INGNE/ AIN(10,10)
COMMON /RENE/ GMIN, HTTE, QTTE, QCH, DCH, DCHR, DCHRT, CCHI, TTEST,

1 GTTEST
DATA AIN /1000.0/, GMIN /1.E-10/
DATA CTMXX /500.0/, DMT1 /TFT/ RNM /D L / DTX1 /30.01/, CXMAX /1.E5/
DATA BZEROD /0/, READ5, READ6 /2*FALSE/, AAMS, AIMOD /2*TRUE/
DATA icase, NELS, INT, IMIN, OMIN, ISW7B /7*0/
DATA NS1 /4/
DATA GMIN /1.E-10/, HTTE /0.01/, QTTE /0.05/, GTTE /0.1/
DATA TTEST /0.05/, QTTE /0.1/, DCH /1.E-4/, DCHRT /1.E-4/
DATA CCHR /0.1/
DATA SAVE /3.0/, LEWIS /1/
DATA ANGLE /10*0/, RADLE /5*0/, NANGLE /0/
DATA NWALLE /0/, WX1, DXWX /2*1/, WX1 /20*1.E30/, NWX /0/
DATA TWEDGE /300.0/, WK /1.333/, ISW7B /0/
DATA NMODT /20/, IGAS /1/
DATA THETA, SN, XS, FRPREF, HR, REFP, QWDOT, TAUW /13*0/
DATA ITPOT, DATAPE /5*0/
DATA CCHR /1.E-05/, CATFAC /1.E-05/, TMODEL /3000.0/
DATA XMODP1 /1.E-20/
DATA TWALL, TPLATE /2*300.0/, IGAS, NOZZO, ICHANO /3*11111/
DATA KDAT /1/
DATA JOIN /1/
DATA CTMXX /500.0/, TST /0.0/
DATA NPREF /2*0/, NRPFLS /1/
DATA EP1 /2HE-1.5, 9597E-4/
DATA EP3 /2HEC-0.026/
DATA EP4 /1MC-120.11/
DATA EP5 /1H, 0.047/
DATA EP6 /1HO, 1.6/
DATA EP7 /10HAR, 39.948/
DATA SP1 /2HE-1.1, 1.1, 1.2*0., 1.2*0., 6.0*0., 1., 14.9276, 0.1, 0.0/
DATA SP2 /1H1N, 1.5, 1.0, 0., 1.0, 0., 6.0, 11.25, 1., 2944,
DATA SP3 /1H0, 1.6, 1.0, 0., 1.0, 0., 6.0, 58990, 1., 4938, 0.0/
DATA SP4 /2HAR, 1.7, 1.0, 0., 1.0, 0., 6.0, 1.8663, 0.1, 0.0/
DATA SP5 /2HN2, 1.5, 1.0, 0., 2.0, 0., 3.451483, 0.88332E-4,
DATA SP6 /2HO2, 1.6, 1.0, 0., 2.0, 0., 3.24973, 0.963449E-4,
DATA SP7 /2HNO2, 1.5, 1.0, 0., 1.0, 0., 3.756216, 2.083961E-4,
DATA SP8 /2HNO3, 1.5, 1.0, 0., 1.0, 0., 3.756216, 2.083961E-4,
DATA SP28 /4HAR2+2e1+1.71+1.0+1.2+6.0+337040+12+3.597+ AR2+  
1 115.11.0.2+9.0.10.0.3*0/  
C AIR-1 IS HIGH-TEMPERATURE AIR MODEL  
DATA GP1 /5HAR1+1.3+1.1+1.2+1.5+1.61  
1 2177.8*0+11.5+1.6+1.2+31.71  
2 8+1.91+1.1+1.1+1.2+1.91+1.0+1.2+1.3+1.41+51.6+1.7+1.81  
3 9+1.10+1.11+1.1+1.2+1.41+51.16+1.17+1.8+1.19+1.20+1  
4 2+1.22+1.23+1.2+1.25+1.380+5+1.6+1.81+1.91+1.1  
5 5+1.0+1.1  
C AIR-2 IS LOW-TEMP. MODEL WITH NO+ THE SOLE ION SPECIES  
DATA GP2 /5HAR2=2.3+1.71+1.1+1.1+1.1+1.5+1.6+1.70+78823  
1 2177.8*0+11.1+1.6+1.2+31.71  
2 8+1.13+1.2+1.3+1.41+51.6+1.7+1.8+1.9+1.10+1+11+1.12+1  
3 131.20+1.20+1.4+51.0+1+131.5+1.0+1  
C NONEQUILIBRIUM MODEL FOR ARGON  
DATA GP3 /5HARON2+1.6+1.17+1.2+1.17+1.8+1.9+1.14+1  
1 26+1.27+1.2+1.28+1.14+76+1.7+1.76+1.79+1+1.80+1.8+1.82+1  
2 83+1.64+1.65+1.66+1.67+1.58+1.99+1.90+1.91+1.91+1.47+0  
3 4+1.90+1.12+1.41+1.2+1.21  
C NONEQUILIBRIUM MODEL FOR HELIUM  
DATA GP4 /5HELIUM2+1.71+1.1+1.2+1.11+3.1+8+0+1.9+0+1.1+1.161  
21+1.22+1.24+1.19+1.23+1.13+1.0+35+1.36+1.37+1.38+1.39+1.40+1  
4+1.42+1.43+1.44+1.45+1.46+1.47+1.48+1.49+1.50+1.51+1.52+1  
C PLANETARY ATMOSPHERE MODEL - MOLE FRACS 0.05 C02, 0.05 N2  
DATA GP5 /5HCONARON2+1.91+1.48+1.6+1+1.41+5+61.7+1.5+0  
1 0.75+0.24+0.057+0.1+1.13+1.31+5+1.6+1.21+31.7+1.14+1  
2 15+1.17+1.8+1.9+1.10+1.11+1.12+1.18+1.20+1.25+1.10  
3 14+1.21+1.3+1.4+1.5+1+1.11+1.12+1.13+1.14+1  
4 15+1.16+1.17+1.18+1.19+1.20+1.21+1.22+1.23+1.24+1.25+1.26+1  
5 5+1.55+1.56+1.57+1.58+1.59+1.60+1.61+1.62+1.63+1.64+1.65+1  
6 6+1.67+1.68+1.69+1.70+1.71+1.72+1.73+1.74+1+1.75+1.16+0  
7 13+1.45+1.7+0+3+13+1.14+1.0+21  
C PLANETARY ATMOSPHERE TEMPERATURES TO 7000 DEG. K  
DATA GP6 /5HCONARON2+1.41+1.33+1.31+1.14+1+1.81+1.51+1.6+1.7+1.5+0  
1 75+2+0.57+0.1+1.41+1.31+5+1.6+1.21+31.7+1.14+1.15+1  
2 17+1.8+1.18+1.25+1.6+0+1+1.21+31.4+1.51+1.6+1.7+1.8+1.9+1  
3 13+1.11+1.1+1.2+1.19+1.20+1.26+1.54+1.55+1.56+1.57+1.58+1.59+1  
4 6+1.61+1.62+1.63+1.64+1.65+1.66+1.67+1.68+1.69+1.70+1.71+1  
5 3+1+1.3+1.4+1.5+1.7+0.3+31.3+1+1.14+1.10+21  
DATA RP1 /5+6E14-1.1+1.17980+1+1+1+1+1.6+1.2+0+31.2+0+1.2+0+1.1  
1 2+1.2+1.7+1+1.13+1+1.14+1.15+1.17+1.13+0  
2 1.2+1.6+1.1+1.6+1.2+1.3+1.6+1.2+1.0+31.6+1.0+2  
3 2+1.6+1.0+1.0+1.0+1.0+1.0  
C PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K  
DATA RP2 /5+6E15-1+1.17980+0+2+1.2+1.6+1.3+1.0+31.2+0+1.1  
1 0.3+2+0.0+1+0+10+0/  
DATA RP3 /3+2E15-1+1+1.17980+0+1+1+1+1.2+6+1.2+0+31.6+1.0+2  
1 2+1+1+1.0+1.0+1.0+1.0  
C PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K  
DATA RP4 /7+2E14-1+1.17980+0+2+1.2+1.6+1.5+0+31.5+1.0  
1 1+1+1+1.0+1.0+1.0+1.0  
C PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K  
DATA RP5 /7+2E15-0+1+1.17980+0+1+1+1+1.5+1.2+0+2+1.2+0+1  
1 1+1+1+1.0+1.0+1.0  
C PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K  
DATA RP6 /4+1E16-1+1.17980+0+2+1+1+1.5+1.2+1+2+1.2+0+1  
1 1+1+1+1.0+1.0+1.0  
C PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K  
DATA RP7 /7+2E15-0+1+1.17980+0+1+1+1.5+1.2+0+2+1.5+1.0+2  
1 2+1+1+1.0+1.0+1.0  
C PLANETARY ATMOSPHERE MODEL FOR TEMPERATURES TO 7000 DEG. K  
DATA RP8 /3+9E14-1+1+1.17980+0+1+1+1+1.2+7+1.2+0+2+1.3+1.0+1  
1 1+1+1+1.0+1.0+1.0
<table>
<thead>
<tr>
<th>DATA</th>
<th>TNE/HE</th>
<th>TNE/AR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DCA 0.75-INCH NOZZLE</th>
<th>NOZZ 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA ZP 1 / 0.95250</td>
<td>0.53974</td>
</tr>
<tr>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>2</td>
<td>-5.68490</td>
</tr>
<tr>
<td>3</td>
<td>3.29700</td>
</tr>
<tr>
<td>4</td>
<td>-12.78253</td>
</tr>
<tr>
<td>5</td>
<td>1.091731</td>
</tr>
<tr>
<td>6</td>
<td>-0.317500</td>
</tr>
<tr>
<td>7</td>
<td>2.225900</td>
</tr>
<tr>
<td>8</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DCA 1.5-INCH NOZZLE</th>
<th>NOZZ 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA ZP 2 / 1.90500</td>
<td>0.14400</td>
</tr>
<tr>
<td>1</td>
<td>4.1</td>
</tr>
<tr>
<td>2</td>
<td>-9.43149</td>
</tr>
<tr>
<td>3</td>
<td>0.0</td>
</tr>
<tr>
<td>4</td>
<td>-22.73861</td>
</tr>
<tr>
<td>5</td>
<td>1.904823</td>
</tr>
<tr>
<td>6</td>
<td>2.031999</td>
</tr>
<tr>
<td>7</td>
<td>0.0</td>
</tr>
<tr>
<td>8</td>
<td>0.0</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>MRA 2.25-INCH NOZZLE</th>
<th>NOZZ 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA ZP 1 / 2.85750</td>
<td>2.71782</td>
</tr>
<tr>
<td>1</td>
<td>6.1</td>
</tr>
<tr>
<td>2</td>
<td>-2.65933</td>
</tr>
<tr>
<td>3</td>
<td>0.032870</td>
</tr>
<tr>
<td>4</td>
<td>5.171749</td>
</tr>
<tr>
<td>5</td>
<td>1.253495</td>
</tr>
</tbody>
</table>
DATA TL35/100*1/
DATA TL36/
11.1960E 00.12451E 00.1290CE 00.12986E 00.12865E 00.12665E 00.12565E 00.1255E 00.1255E
21.2455E 00.12253E 00.12078E 00.11919E 00.11978E 00.11878E 00.1172E 00.11496E 00.11496E
31.1366E 00.11270E 00.11197E 00.11080E 00.11016E 00.10980E 00.10980E 00.10980E 00.10980E
41.0958E 00.10935E 00.10925E 00.10925E 00.10925E 00.10925E 00.10925E 00.10925E 00.10925E
51.0923E 00.10927E 00.10930E 00.10930E 00.10930E 00.10930E 00.10930E 00.10930E 00.10930E
61.0943E 00.10944E 00.10944E 00.10944E 00.10944E 00.10944E 00.10944E 00.10944E 00.10944E
71.0957E 00.63*1/
DATA TL40/ 95.0.0.1.2133.1.0.1.165.0.0 /
END
SUBROUTINE AESOLN (X)

COMMON /RL/ DELBL(2),RLT(2),XZERO,TWALL,CPWALL,VSURF,DIAM(2),

COMMON /AREA/ ATP1(11),ATP2(8),RTHCM(2),NSECT(2),iSHAPE(2),NFROFL(2),PHFL,SNBL

COMMON /AEGEOM/ SQRTA,Sl,SE

COMMON /ERROR/ ERA

COMMON /NEO/ OMUST(2),DDELBL(2)

IF XEQ.0) GO TO 10

SORTAE= (SQRTA-DELRL(I))/OMOST(I)

S1= SQRTA*2

S2=(S2/SORTA-2.*DELBL(I)/OMOST(I)*SORTAE)

GO TO 40

10 S1= (SI-DELBL(I))/OMOST(I)

S2=(S2-DELBL(I))/OMOST(I)

GO TO 40

20 CALL GEOMAR (X,DX.DZ,DY,DYDX,DYDZ)

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/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12,2), NPRFLS, NBL
COMMON /NEO/ CMDST(2), DELNEL(2)
COMMON /ERROR/ ERR
COMMON /BL/ DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SW, RO, JDIM, IPCINT
DIMENSION DEL(2), DELP(2), DSTM(2)
DATA RNAME /SHAGSOLN/

IF (NPRFLS, EG, 2) GO TO 20
NOT A CHANNEL
IF (JDIM EQ 0) GO TO 10

C TWO-DIMENSIONAL FLOW
AG = DEL(1) * CMDST(1) * AE
GO TO 100

C AXISYMMETRIC FLO\nSORTA = DEL(1) * CMDST(1) * SORT(AE)
AG = SORTA ** 2
GO TO 100

C CHANNEL
DO 30 L = 1, 2
DELPH(L) = RO ** (1. - CMDST(L))
30 FIX = FALSE
Y2ZC = RTHCM(1) * RTHCM(2)
AEC = (1. - DSTM(1) / RTHCM(1)) * (1. - DSTM(2) / RTHCM(2)) * AE - DELPH(1) * DELPH(2)

1 / Y2ZC
A = AE
ICOUNT = 0
CALL FINDX (A, UPDOWN, X)
IF (ERR) RETURN
ICOUNT = ICOUNT + 1
IF (ICOUNT GT 20) GO TO 90
CALL GMAR2 (X, Y, Z)
IF (ICOUNT GT 1) GO TO 70
A = AEC * (Y * DELPH(2) + Z * DELPH(1)) / Y2ZC
X = X
PQ = A - AE
IF (A GE 1.) GO TO 50
X = 0
GO TO 60

70 CALL GEOMAR (X, A, DA)
IF (ERR) RETURN
F = AEC * (Y * DELPH(2) + Z * DELPH(1)) / Y2ZC - A
IF (ABS(F) / ALE ** 1 ** E - 5) GO TO 120

AG 1
AG 2
AG 3
AG 4
AG 5
AG 6
AG 7
AG 8
AG 9
AG 10
AG 11
AG 12
AG 13
AG 14
AG 15
AG 16
AG 17
AG 18
AG 19
AG 20
AG 21
AG 22
AG 23
AG 24
AG 25
AG 26
AG 27
AG 28
AG 29
AG 30
AG 31
AG 32
AG 33
AG 34
AG 35
AG 36
AG 37
AG 38
AG 39
AG 40
AG 41
AG 42
AG 43
AG 44
AG 45
AG 46
AG 47
AG 48
AG 49
AG 50
AG 51
AG 52
AG 53
AG 54
AG 55

END
IF (F EQ FO) GO TO 120
XL=X
X=X-F*(X-XO)/(F-FO)
XO=XL
FC=F
IF (X*UPDOWNGE0) GO TO 60
SIGN OF X IS INCONSISTENT WITH UPDOWN
IF (FIX) GO TO 90
IF (AE GT 100) GO TO 90
TRUBLE IS DIAGNOSED AS DUE TO SONIC POINT TOO FAR DOWNSTREAM
OF GEOMETRIC THROAT. TRY FIX -- RESET DISPLACEMENT THICKNESSES
TO THEIR VALUES AT THE THROAT.
DO 80 L=1,2
DELP(L)=DSTP(L)
WRITE (6,140)
GO TO 40
WRITE (6,150) AE,DEL,UPDOWN,X
NAMELIST /AGDMP/ OMOST,RO,DSTP,DELP,Y0ZO,RTHCM,AEC,A AG,XO,F,FO,
Y,Z,ICOUNT
WRITE (6,AGDMP)
CALL DUMP (RNAME)
GO TO 130
IF (AGGT10) GO TO 110
AG=1.
X=0.
GO TO 130
CALL FINDX (AG,UPDOWN,X)
GO TO 130
AG=A
RETURN
C
C
FORMAT (29H0+++++ FIX REQUIRED IN AGSOLN)
FORMAT (33H0CONVERGENCE FAILURE IN AGSOLN,9X,3HAE=1PE12.5,8X,4HE=1PE12.5)
END
SUBROUTINE AXFIT
REAL ACOM(30), ELMNT(10), HP(20)
C DETERMINES AXIAL POSITION DURING PERTURBATION SOLUTION
DOUBLE PRECISION AA, CAPX, 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, CMA, CRA, CRP
2 CRRB, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMAX
3 CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT1
4 DELT2, DELTAX, DLOGR, DT, ENT, FLUX, HDELX, PCT
5 SPECTST, PREX, PRESA, PRESB, PRESTH, PRHO, RHAP, RHOB
6 RHOBAR, RMHC, RMOP, RMPL, RMTH, ROBAR, ROBAP, SCPL, SDT
7 SEN, SHPG, SC, SL, SLA, SM, SU, SUZ, SUMG
8 TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA
COMMON EE(4), BET(20), BLBK(31), CAI(64), CAPXTH(20)
1 CCCI(20), CEACT(64), CGL(20), CGLA(20), CHD(20)
2 CEXC(64), CEXN(64), CEXD(64), CEXN(64), CHI(64)
3 GJ(20), PERTGJ(20), PGJ(20), PI(64), PDEH(20)
4 Q0(64), SAJ(20), SDCH(64), SNOT(20), SHJ(20), SHJA(20)
5 SKIL(20), SS(20), TH(20), TFA(20), TFD(20), TFC(20)
6 TFD(20), TPE(30), TKF(20), TKE(20), XMJAT(20), XNUL(64)
COMMON RETA(64, 20), FLJ(10, 20), GFLJ(10, 20)
1 XNUIJ(64, 20), XNUIJP(64, 20)
COMMON IC, IM, INEQ, INDV, IP, IRUN, ISC, ISCP1
1 ISMC, ISME, ISMR, ISSI, ISSP1, ISSP2, ISSP3, ISSP4
2 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B
3 ISW5A, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, MI, NFT
4 NIT, NNN, NS, NOT, NTEST
COMMON IGJ(30), IGJ(20), ITB(5), KUR(64, 20), LPIJ(20, 10)
COMMON ACOM, ELMNT, HP
COMMON VBL, DELCV(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2)
1 SW0, RO, JDIM, IPCINT
IF (CT-CTMAX) 20, 20, 10
10 UPDOWN=1
GO TO 30
20 UPDOWN=1
GO TO 30
30 IF (ISW3B, NE, 0, AND, IUPD, EQ, 0) GO TO 40
CALL FINDX (AFNTS, UPDOWN, AG, CX)
GO TO 30
40 CALL AGSOLN (AFNTS, DELBL, UPDOWN, AG, CX)
RETURN
END
SUBROUTINE AXSF игры (LL, NN)
SUBROUTINE AXSF игрыadds data for unspecified species pairs to step
LL of the edited cross section data. Species to be considered are:
GIVEN IN NUMERICAL ORDER IN THE FIRST NN ENTRIES OF THE ARRAY I.
COMMON /TRAN1/ V(3,20,20), ZM2(20)
COMMON /TRANS2/ KQ(100), NQ(100), ISEQ(100), NN0, I(50)
COMMON /TRANS7/ V(400), KQ(100), NQ(100), I(400), JQ(400), NKQ
IF (NN LE 0) GO TO 40
LQ1 = NQ(LL) + 1
N = 0
DO 20 IV = 1, NN
II = I(IV)
DO 20 JV = IV, NN
JJ = J(JV)
IF (Q11, II, JJ) NE 0) GO TO 20
Q(1, II, JJ) = 1
N = N + 1
LQ = 400
DO 10 L = L01, 399
LQ = LQ - 1
10 JQ(LQ) = JQ(LQ - 1)
JQ(LQ) = II
JQ(LQ) = JJ
CONTINUE
DO 30 L = LL, NKQ
30 NQ(L) = NQ(L) + N
40 RETURN
END
COMMON /SS/ CAS,US
COMMON /SWITCH/ XPB,DXPB,W
EQUIVALENCE (XP,XPU), (DXP,DXPB)
COMMON /DRW/ SHJDGJ
COMMON /TNCE/ SUMGH,SCPGH,ODPR,ODPE
COMMON /BLRAD/ YQ70
COMMON /MEDPE/ MODLP2
COMMON /CLUDS/ CGMW,OPJ(10),IJC(10),NCS
COMMON /TNONE/ CTE,DTE,EPR,EPAIR(2,25),INT,ITR(25),KTF(25)
1       KTR(25),ICH(2C),IPA(25)
DATA ONE /1/=1, ZERO /0/=0
NAMELIST /UMP/ IPOINT,IX,IPR,PH,UM,AM,HE,HE,HR
1       HREF,HW,TE,TFRE,CSB,DXZ,YZ,DYDX,ZVSC,SCVR,VCSC
2       PR,PRREW,PRW,OMEGA,OMEG,DM,DMX,DMU,DM,RO
3       SW,NBL,ITYPE,REF,BL,ENFR,SN,SN,THTA1,THTA,THTA,HTR
4       DELBL,DELBL,IL,RAF,TAUW,CWDT,XTEST,GAMMAE,WW,AVCN,B,KE1
5       OLDMF,TRNCALC,TEP,TEP,REM,REM,RO,F0,F2

DELBL=DISPLACEMENT THICKNESS

Theta=MOENTUM THICKNESS

THE BOUNDARY LAYER THICKNESSES DELBL, Theta, Theta1 ARE ALL
NODIMENSIONALIZED USING THE GEOMETRIC THROAT RADIUS RO. THEY ARE CONVERTED TO INCH UNITS FOR OUTPUT, IN SUBROUTINE OUTI.

IF (X. GT. XP OR IPOINT.EQ.1) GO TO 10
IF (X. EQ. CXH) GO TO 550
WRITE (6,560) X,XP
WRITE (6,561)
GO TO 550
10       DO 540 L=1,NPRFLS
IF (IPOINT.EQ.1 OR L.EQ.2) GO TO 1000
540       SN=1+SN
B=2.141*1.28*SWP1+0.93*(SWP1**2)**2
3=M1=-B-1.
1000      EKI AND B ARE THE COEFFICIENTS IN THE CAHM-RESHOTKO LINEAR FIT TO
THE MOMENTUM PARAMETER (CAPA,?) AS A FUNCTION OF THE CORRELATION NUMBER SN (SMALL N). THE FORMULA IS N=EKI+B*SN.
SVFAC=SWP1*(7.3*3.4*SWP1)
SWFAC2=1.4*5*SWP1**0.9
HCON=CRP/CMA
WW=PWALL*TWALL
TEQ=0.
550       TREFP=90.
600       TQ=CHAPC*HCON
DO 20 1=1,ISS
XSAVE(I)=SAVEC(I)
DO 20 1=1,ISS
DO 30 K=1,NCS
IF (IJC(1) .GT. NE.1) GO TO 30
SAVEC(I)=OPJ(I)
DO 30 1=1,ISS
30       CONTINUE
SAVEC(I)=0.
40       CONTINUE
CALL TRANSP (TWALL,PRESA)
PRW=PR
VISCV=VISC
RWM*VISCV*PRESA*CGMW/8*05/TWALL/RHAP
DO 50 I=1,15
S=VISCW*PRESA*CGMW/~~~O~~TWALL~RHAP
D'=3
C=I*ISS
S4VEC(I)=XSAVE(I)
IF (NPfiFLSeEOeL)
GO TO 60
YDZ0=XTHCM(1)*RTHCM(2)
ITYPE=3
GO TO 80
ITYPE=1
ITYPE IS A FLAG FOR TYPE OF NOZZLE GEOMETRY
ITYPE=1 TWO-DIMENSIONAL
ITYPE=2 AXISYMMETRIC
ITYPE=3 RECTANGULAR CHANNEL
AVCON=W/R0
R0 IS SET IN MAIN
COUPLD=FALSe*
ISMD=30
DDELBL(1)=0.
DDELBL(2)=6.
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
L.INTI=PLINT(L)
XI=X(L)
110 IF (L.EQ.2) GO TO 220
TE=CTAP*T
DO 120 I=1,155
IF (AMS(XTEST(I)-SAVEC(I))*GT*40.01) GO TO 130
120 CONTINUE
CLDFM=F*TRUE*
TRCALC=F*FALSE*
GO TO 150
130 CLDFM=F*FALSE*
DO 140 I=1,155
140 XTEST(I)=SAVEC(I)
GO TO 160
150 IF (ABS(TC-TEP)/TE*L.T.*0.005) GO TO 180
160 CALL TRANSP (TE,P,PRESA)
TEP=TE
IF (EFR) RETURN
IF (IPOINT.L.E.1) PRREF=PR
DO 170 I=1,4
170 TRPSV(I)=TRP(I)
180 ME=HCON*H
PRREF=PR**0.56
HR=ME+PRW*(H0-HE)
HRE=RECOVERY ENTHALPY
HREF=C***(HF+Hw)+0.22*(HR-HE)
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(1)=1./SORT(1.*(RO*DAX)**2)
GO TO 230

200 AJ=AG
COSB(1)=1./SORT(1.*0.25*(RO*DAX)**2/AG)
GO TO 230

210 CALL GMAR3 (X,DYDZ(1),DYDZ(2),YZ(1),YZ(2))

220 LPR=LPRIME(L)
AJ=(YZ(LPR)**2
COSB(COSB=COHSCN THE ANGLE BETWEEN THE NOZZLE WALL AND THE AXIS OR CENTER PLANE)

230 IF (L.EQ.0) GO TO 270
UPRIME=U*US
IF (L<500 AND ABS(TREF-TRFFP)/TREF*LT.0.005) GO TO 240
CALL TRANSP (TREF,P*PRESA)

IF (ERR) RETURN
TREF=P*TREF
TRCALC=TRUE
PRREF=PR
VISCR=VISL

240 IF (*NOT,TRCALC) GO TO 250
VR=VISCR/VISCW
TR=TREF/TWALL
OMEGA=ALOG(VR)/ALOG(TR)

IF (IPOINT<NE+1) GO TO 250
EK=0.38-0.276*PREF*1.*CMega+EXP(-6.67*SWP)

IF (EK>GE*0.01) GO TO 250
WRITE (6,530)
WRITE (6,530)

530 EK=0.01

250 DO 260 I=1,4
260 TRP(I)=TRSV(1)
AM=UP./ME/AS
ORDIN(L)=AJ*AM**BM1*AS*RH/T

IF (ITYPE=EQ.2) R=SORT(AG)

270 C R=RATIO OF LOCAL NOZZLE RADIUS TO THROAT RADIUS
IF (IPOINT<NE+1) GO TO 290

FIRST POINT
DX=(X-XZERO)/10.
DX0=DX**2
XX=XZERO
X1(I)=0.
PL1NT(I)=0.
CC=AM*AG
CALL RADIUS (ITYPE,XZERO,R1,AG1,AGJ,L)
AXI=CC/AG1
ORDI=AGJ*AM1**8M1
GO 280 I=1,10
XX=XX+DX
CALL RADIUS (ATYPE+XX*R2*AGL*AGJ*L)
DXI=SQRT((DXS0+(R2-R1)**2)
XI(L)=XI(L)+DXI
AM1=CC/AG1
ORD2=AGI*AM1**BM1
BLINT(L)=BLINT(L)+0.5*(ORD1+ORD2)*DXI
ORD1=ORD2

280 R1=R2
BLINT(L)=BLINT(L)*AS
DLH(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 BLA
C 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(1)*CCPJ(1)*DTE+SCPG*DT
GO TO 310
300 DUDX=SCPG*DT
310 DUDX=CMA/J*(DUDX+SHJMGJ)
IF (NT.EQ.2) DUDX=DUDX+DCHA/SU
DLB(L)=DUDX/U-0.5*DT/T
GO TO 360
320 IF (IPQNT.GT.2) GO TO 340
330 DMDX=(AM-AMP)/DX
GO TO 350
C CALCULATE DERIVATIVE FROM QUADRATIC FIT
340 IF (DX*LF.0) GO TO 370
DX2=DX+DXP
XPXP=X+XP
DMDX=(AM-AMP)/DX
C2=(DMDX-(AM-AMPP)/DX2)/DXP
C1=DMDX-C2*XPXP
DMDX=C1+2.*C2*X
IF (DMDX+L+0.4) GO TO 330
350 DLM(L)=DMDX/AM
C TRAPEZOIDAL RULE INTEGRATION
360 DX=DX/COS(L)
XI(L)=XI(L)+DXI
BLINT(L)=BLINT(L)+0.5*(ORDIN(L)+ORDINP(L))*DXI
370 DLM(L)=DLM(L)*COSB(L)
REH=RH*HAP*URPM*RO/VISC
C RER=REYNOLDS NUMBER BASED ON THROAT RADIUS AND FREE-STREAM CONDITIONS.
C ENPM=BLINT(L)/ORDIN(L)/RO*EK1
RM=RW*WP/VISC/RH
THETA1=QTET(ENPM/RER*RMR)
GO TO (380,390,380), ITYPE
380 THETA(L)=THETA1
GO TO 400
C TRANSVERSE CURVATURE CORRECTION FOR AXISYMMETRIC NOZZLES
390 SQT=SQRT(1.-2.*COSB(L)*THETA1/R)
THETA(L) = R/COSB(L)*(1 - SOT)

IF (not FINAL AND, IPONI = NE) GO TO 430

SN(L) = ENPRIM*S00*DLM(L)
IF (IPONI = NE) GO TO 410
XSN(L) = SN(L)
GO TO 420

WW = EXP(-AVCON*DX/AG)
XSN(L) = (1 - WW)*SN(L) + WW*XSN(L)

XSNV = AMIN1(XSN(L), ZERO)
ALPR = ALOG(PRREF)
GAMMAE = AS**2/(2.314*CM*TE)
IF (GAMMAE > 1) GAMMAE = 1.0

SIGH = 1.0/(1 + 2.0/GAMMAE**2)
HTR(L) = 0.75*(EXP4.6*XSNV)**1 + (3.0 - 15.6*XSNV)*SWP1 - 1.0
HTRP1(L) = (HTR(L) + 1.0)*(SWP1**(-0.4*ALPR) - 0.47*SIGH*ALPR)

IF (not FINAL AND, IPONI = NE) GO TO 470
IF (COUPLE = 1) GO TO 440
IF (IUPD = 0) GO TO 470
COUPLO = 1.0

DELBL(L) = H*THETA(L)
IF (IPONI = LE = 1.0 OR, not FINAL) GO TO 470
IF (COUPLE) GO TO 440
IF (IUPD = 0) GO TO 470
COUPLO = 1.0

DDDELBL(L) = (DELRL(L) - DELBL(L)) / (X - XP)

IF (ISMDOEQ = 1.0 OR, LEQ = 2.0) GO TO 450
WD = 1.0 / ISMDI
GO TO 460

IF (L = 2.0) GO TO 460
WD = 0.5*AMINI(DIX/DX)**2

DDDDLRL(L) = WD*(DDDELBL(L) + (1.0 - WD)*DDBLP(L))

IF (ISMDOEQ = 1.0 OR, LE = 1.0) ISMDO = 1.0
GO TO 470

CONT, NUF
IF (FINAL) GO TO 480
BLINT(L) = BLINTI

IF (FINAL) GO TO 480
BLINT(L) = BLINTI

IF (FINAL) GO TO 480
BLINT(L) = BLINTI

IF (FINAL) GO TO 480
BLINT(L) = BLINTI
$\text{TAWU}(L) = 0.0C209 \times 10^X \times \text{MR} \times \text{VISC} \times \text{UPRIME} / \text{THEETA}(L) / 0$

$\text{RA} = \text{SW} \times 10^X \times (1 - 0.86 \times \text{OEMEGA}) \times \text{SW} \times 10^X$ + (1 - \text{PRREF}) \times 10^X$

$11.55 \times (1.2 - \text{SW} \times 10^X \times 0.29 \times \text{SW} \times 10^X \times 0.38 \times \text{SIGH} / \text{OMEGA})$

$\text{R} = 0.94 - 1.31 \times \text{SW} \times 10^X \times 0.486 - \text{ALPR} \times (0.62 \times \text{SW} \times 10^X \times (3.5 \times \text{DM} - 0.49 - 3.61 \times \text{SW} \times 10^X)$

$\text{SIGH}$

$\text{RAF} = \text{PRREF} \times ((1 - 0.35) \times (1 - \text{SN} \times 10^X) / (R\text{A} - R\text{B} \times \text{SN}(L)) )$

$\text{Q} $ $\text{DOT} (L) = 1760 \times \text{TAWU} \times (1 - \text{HW} \times \text{PRW} / \text{RAF}$

$\text{DDEL} \times 10^X = \text{DDEL} \times (L) \times \text{DDEL} \times (L) \times \text{DDEL} \times (L) \times \text{DDEL} \times (L)$

$\text{IF} (\text{L} \times \text{LT} \times \text{NPRFLS}) \text{GO TO 540}$

$\text{IF} (\text{L} \times \text{LT} \times \text{NPRFLS}) \text{GO TO 540}$

$\text{AMPP} = \text{AMP}$

$\text{AMP} = \text{AMP}$

$\text{X} \times X = \text{X} \times \text{X}$

$\text{DX} \times \text{DX} = \text{DX} \times \text{DX}$

$\text{IF} (\text{NOT} \times \text{MODLPT}) \text{DX} = \text{DX}$

$\text{REP} = \text{REP} / \text{R} \times 30 \times 48$

$\text{CONTINUE}$

$\text{IF} (\text{ISW} \times 10^X \times \text{GT} \times 0) \text{WRITE} (6, \text{DMP})$

$\text{IF} (\text{ISW} \times 10^X \times \text{LT} \times 0) \text{WRITE} (6, 570) \times \text{X}, \text{IPOINT}, \text{ISMD}, \text{WD}, (\text{DDEL} \times (L) \times \text{J} = 1 \times \text{NPRFLS})$

$\text{LS}$

$\text{RETURN}$

$\text{FORMAT} (36 \text{HO} \text{ZERO} \text{OR} \text{NEGATIVE} \text{STEP} \text{IN} \text{BLAYER}, X = 1 \text{PE} 15 \times 8, 10X, 3 \text{HXP} = 1 \text{IE} 15 \times 9)$

$\text{FORMAT} (12 \text{H} \text{BLAYER}, X = 1 \text{F} 10 \times 5 \times 7 \text{HIPOINT} = 14 \times 5 \times 5, \text{ISMD} = 12 \times 5 \times 3 \text{HWBLA} 356$

$\text{ID} = 1 \text{F} 6 \times 4 \times 5 \times 7 \text{HD} \times \text{ELBL} = 2 \text{F} 15 \times 6$

$\text{FORMAT} (56 \text{HO} \text{****} \text{CALCULATED EK1 LESS THAN 0.01} \text{EK1 RESET TO 0.01})$

$\text{END}$
SUBROUTINE BLCALL (FINAL)
CALLING ROUTINE FOR BOUNDARY LAYER IN NONEQUILIBRUM CALCULATIONS

LOGICAL FINAL

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, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO

1 C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP

2 CRRA, CRST, CT, CTAP, CTB, CTC, CTMAX, CTMXN

3 CTP, CTPL, CTT, CX, CXB, CMXMAX, DATEST, DBTEST, DELT1

4 DELT2, DELTAX, DLOGA, DLOGB, DT, ENT, FLUX, HDELX, PCT

5 PCTEST, PRES, PRESA, PRESB, PRESTH, PRHC, RHAP, RHO, RHOB

6 RHOBAR, RHOD, RHOP, RP, RTH, ROBAR, ROBARP, SCPG, SDT

7 SEN, SHPG, SC, SL, SL64, SN, SU, SU2, SUMG

8 TEST, TESTB, PRINT, STOP, UP, ZP, ZPA

COMMON BE(64), BLK(31), CAI(64), CAPXTH(20)

1 CCPJ(20), CEACT(64), CGI(20), CMU(20), CHI(64), CHII(20)

2 CLNMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJ(20), GJA(20)

3 GJ(20), PERTGJ(20), PGJ(20), PI(64), PICHI(64), OM(20)

4 OU(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20)

5 SKL(20), SS(20), TBA(20), TFA(20), TFB(20), TFC(20)

6 TFD(20), TFE(20), TFK(20), THE(20), XMJAT(20), XNUJ(64)

7 COMX(64,20), ELJ(10,20), GELJ(10,20)

1 XNUJ(64,20), XNUJ(64,20)

COMMON IC, IM, INEQ, INEQV, IR, IRUN, IS, ISC, ISCPI

1 ISMC, ISMCN, ISR, ISS, ISSR, ISSP1, ISSP2, ISSP3, ISSP4

2 ISP1, ISP2, ISP3, ISP4, ISP5, ISP6

3 ISP7, ISP8, ISP9, ISP10, ISP11, ISP12

4 NIT, NNN, NNS, NOS, NOT, NTEST

COMMON IGJ(20), IGMIN(20,10), IGMIN(20,10)

1 IGMIN(20,10), IGMIN(20,10)

COMMON ACOM, ELMNT, HP

1 DIMENSION AAA(22,24), BTA(64,20), CAPO(31), CCI(20), DGJ(20), GJ(20)

2 SBPLC(20)

COMMON I(20), SDGJ(20), SHJAP(20), THEV(20)

1 EQUVALENCE (AA(1,1), AA(1,1), (BETA(1,1), BTA(1,1)),

2 (BLK(1,1), CAPQ(1,1), (CAPXTH(1,1), CCI(1,1)), (GJA(1,1), DGJ(1,1)),

3 (SHJA(1,1), SHJAP(1,1), (THE(1,1), THEV(1,1))

COMMON /TEMPC/ SAVFC(20)

1 IF (INEQ*EQ.0) GO TO 20

2 DO 20 J=1,ISS

3 SAVFC(J)=GJ(J)*CM

4 GO TO 40

5 CT=CT+PCT

6 RHO=RHO+PRHO

7 DO 30 J=1,ISS

8 SAVFC(J)=(GJ(J)+PERTGJ(J))*CM

9 IF (1SW3*EQ.0) GO TO 50

10 CALL BLAYER (FINAL)

11 IF (INEQ*EQ.0) GO TO 60

12 CT=CT-PCT

13 WHO=G4C-PRHO

14 2FTURN

15 END ecc
SUBROUTINE BXSEC(L,MV,I,J)

SUBROUTINE BXSEC DETERMINES THE CORRESPONDENCE BETWEEN THE KQ AND V ARRAYS AND SEARCHES THE V ARRAY FOR REFERENCES TO SPECIES PAIRS.

COMMON /TRANS/ V(400),KQ(100),NG(100),IQ(400),JQ(400),NKQ

COMMON /TRANS/ NV(15),N

IF (L*NE.0) GO TO 40

MV=1

MADD=0

L=L+1

K=KQ(L)

IF (K*EQ.9) GO TO 30

IF (K*NE.12) GO TO 50

MADD=4

MADD=MADD-4

GO TO 30

MY=MV+UADD

I=V(MY+1)

J=V(MY+2)

RETURN

40

V(MY+1)=I

V(MY+2)=J

IF (MADD*NE.0) GO TO 20

IF (L*LT*NKQ) GO TO 10

L=0

RETURN

END
SUBROUTINE COMM

LOGICAL ERR, FAILED, LOGC, NOREAC, RCALC

REAL ACOM(30), ELMENT(10), HP(20)

DOUBLE PRECISION AA, AAA, CAxK, GJ, CDI

COMMON AA(22, 24), CDI(20, 10), CAPX(20)

COMMON A, AFNX, AFNX, AMACH, AR, ARBA, ARBR, BZERO, CALLR

1 C, CARB, CH, CHA, CLNT, CM, CMA, CMA, CRP

2 CRRB, CRS, CSTA, CT, CTAP, CTB, CT, CMAX, CMAX

3 CPS, CPI, CTT, CX, CXG, CMAX, DATEST, DTTEST, DTET

4 DELT, DELT, DELTA, DLOG, DLOG, HMAX, HMAX, PCT, PCT

5 hPPTEST, HEPAS, PEEP, PEP, PRHO, PRH, PRH, RHOD

6 RMODAR, RHOC, RHOP, RHOP, RHOP, ROBRA, ROBRA, SCPG, SCPG, SCPG

7 SEN, SHPG, SC, SL, SLOA, SM, SU, SU2, SUMG

8 TEST, TEST, TNTINT, TSTOP

9 UP, ZP, ZPA

10 COMMON DE(64), BET(20), BLK(31), CAI(64), CAPXTH(20)

11 CCPJ(20), CEACT(64), CGI(20), GCMU(20), CHI(64), CHII(20)

12 CLNIMC(64), CLNIP(64), CMW(20), ETA(31), ETA(20), ETA(20)

13 GJ(20), PERTGJ(20), PGJ(20), PI(20), PIC(20), QM(20)

14 GJ(20), SJX2(20), SDCH(4), JENT(20), SHJ(20), SHAJ(20)

15 SHJ(20), SS(20), T3(30), TPA(20), TFB(20)

16 TF(20), TE(20), TF(20), TVE(20), XNJAT(20), XNUI(64)

17 COMMON HST(64, 20), ELJ(10, 20), GELJ(10, 20)

18 COMMON IC, IM, INEG, INFQV, IP, IRUN, ISG, ISCP1

19 ISMC, ISMCN, ISR, ISS, ISSN, ISSP1, ISSP2, ISSP3, ISSP4

20 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4, ISW5A, ISW5B

21 ISW6, ISW6, ISW6, IUPC, IZPQ, JJK, LC, M1, NFI1

22 JI, JII, NNN, NOS, NOT, NTEST

23 COMMON IGJ(20), IGJ(30), ITB(3), JUK(64, 20), LPIJ(20, 10)

24 COMMON ACOM, ACOM, ACOM, ACOM, ACOM

25 DIMENSION AAA(22, 24), BTA(64, 20), BAP(31), CCI(20), DGI(20), GJ(20), SBC(20)

26 COMMON /AII(1), AII(1), (HETAT1(1), BTA(1, 1))

27 COMMON (BLK(1), CAPX(1), COMX(1), CII(1)), (GJA(1), GJ(1))

28 COMMON (CAPX(1), GJ(1), SJX2(1), SJX2(1), ISW(1), SDXG(1))

29 COMMON (SHJ(1), SHJAP(1), THEV(1), THEV(1))

30 COMMON /ERROR/ ERR

31 COMMON /ERROR/ ERR

32 COMMON /ERROR/ ERR

33 COMMON /ERROR/ ERR

34 COMMON /ERROR/ ERR

35 COMMON /ERROR/ ERR

36 COMMON /ERROR/ ERR

37 COMMON /ERROR/ ERR

38 COMMON /ERROR/ ERR

39 COMMON /ERROR/ ERR

40 COMMON /ERROR/ ERR

41 COMMON /ERROR/ ERR

42 COMMON /ERROR/ ERR

43 COMMON /ERROR/ ERR

44 COMMON /ERROR/ ERR

45 COMMON /ERROR/ ERR

46 DATA VICE(0), VICE(0), 100/EO, 100/EO, 100/EO

47 DIMENSION ALGJ(20)

48 EQUIVALENCE (ALGJ(1), AA(1, 24))

49 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

50 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

51 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

52 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

53 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

54 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

55 NAMELIST/CCMDNP/ CX, AFNX, DLOG, RHO, DLOG, SUMG, SHPG, RHAP, CRP,

56 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

57 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

58 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

59 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

60 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

61 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

62 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

63 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

64 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

65 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)

66 IF (ISW5, 3) IF (ICOUNT, ICOUNT + 1)
IF (ICYCLE.EQ.90000000) GO TO 30
IF (MOD(ICYCLE,ICYCLE)) 10,20,10
10 T=WSB=0
JJ TO 30
20 IS=WSB=1
WRITE (6,550) ICOUNT
30 CONTINUE
FAILED=FALSE
SUMG=0
SCPGH=0
QDP=0
QDPE=0
DO 70 J=1,ISS
IF (G(J,J)) 80,40,50
40 ALGJ(J)=-1000
GO TO 60
50 ALGJ(J)=ALOG(G(J,J))
GO TO 60
60 IF (J.LT.IJNT) GO TO 70
SUMG=SUMG+G(J,J)
SCPG=SCPG+G(J,J)*CCPJ(J)
70 CONTINUE
SUMG=SUMG
SCPG=SCPG
IF (NT.EQ.1) GO TO 100
SUMG=SUMG+G(J(1))
SCPG=SCPG+G(J(1))*CCPJ(1)
TCP=CTET*CTAP
GO TO 100
WRITE (6,*560)
90 FAILED=TRUE
RETURN
100 IF (INFO) 110,150,110
110 SFN=0
SHPG=0
DO 130 J=1,ISS
IF (G(J,J)) 80,130,120
120 SFN=SFN+G(J,J)*(SENT(J)-ALGJ(J))
130 SFN=SFN+G(J,J)*60
CM=1./SUMG
SHPG=SHPG+G(J,J)*SHJ(J)
CH=CMASK(HPG)
SU2=2.+CH
IF (SU2.GE.0.) GO TO 140
WRITE (6,570)
GO TO 90
140 SU=SGRT(SU2)
150 CALL GECM
IF (ERR) RETURN
RMOP=RH0*RHAP
RHL=ALOG(RMOP)
ZI=ALOG(SL64/SU)+CSTA
Z2F=1.2918*CRP
PCALC=FALSE
C MAIN LOOP OVER THE REACTIONS
DO 460 ,1,150
460 IF (INCERAC(I)) GO TO 450
COMPUTATION OF PICI

LOGC=.FALSE.
IF (NT*EQ.1) GO TO 160
KF=KT'(1)
KR=KTR(1)
GO TO 170
160 KF=1
KR=1
170 IF (KF-3) 180,190,200
180 CJKF=ETA(I)*CLNTD(KF)+CAI(I)-CEACT(I)/(CTD(KF)*CRP)
GO TO 240
190 CJKF=ETA(I)*CLNTD(2)+CAI(I)+ALOG(1.-EXP(-CEACT(I)/(CTD(1)*CRP))
GO TO 240
200 IF (RCALC) GO TO 220
RCALC=.TRUE.
IF (NPRFLS*EQ.2) GO TO 210
CALL GMT (CX,R)
GO TO 220
210 CALL GMAR2 (CX,Y,Z)
R=I./SQR(D(C,Y**2+Z**2))
220 IF (KF*EQ.4) GO TO 230
CJCF=CAI(I)-0.5*ALOG(R)
GO TO 240
230 IPA=IPA(I)
TAU=SPAR*RHOP*GJ(IPAI)*R
CJCF=ETA(I)*CLNTD(2)-ALOG(MAX1(TAU,TAU))
240 CJNP(I)=CJKF+Z1
CJCF=CJCF
CJ=CMC(I)=0.0
T6=0.0
T7=0.0
IF (KR*EQ.0) GO TO 250
Z2=Z2F*CTD(KR)
Z3=Z2*RHOP
CLZ2=ALOG(Z2)
250 D=E=200 J=1.1SS
IF (XNU1(J),J)*EQ.0.0) GO TO 260
T5=T5+XNUI(J)*ALGJ(J)
260 IF (KUR(I),J)*EQ.0) GO TO 270
T7=T7+GJ(J)
270 IF (BETA(I,J)*EQ.0) OR (KR,EQ.0) GO TO 280
CLNMC(IMC(I)=CLNMC(I)+BETA(I,J)*(ALGJ(J)+CLZ3*XMJADT(J,KR))
280 CONTINUE
IF (KR*EQ.0) GO TO 300
IF (CLNMC*LT.20) GO TO 290
LOGC=.TRUE.
GO TO 310
290 CHI(I)=1.0-EXP(CLNMC(I))
GO TO 310
300 CHI(I)=C
310 CJNP(I)=CJNP(I)+(XNU1(I)-1.0)*RPM+T6
IF (Q(Q)) 320,350,320
320 IF (T7) 330,340,330
330 CLTB=F+ALOG(RHOP*T7)
\text{ODPE} = \text{ODPE} + \text{GELAS}

\text{PRES} = \text{PRES} \times \text{RHO} \times \text{CMA} / \text{ROBARA}

\text{SEN} = \text{CRA} \times (\text{SEN} - \log (\text{PRES} \times \text{CM} \times \text{PRESA}) / \text{CM})

\text{CONTINUE}

\text{IF (INEQ} \neq 0 \text{ AND ISW5B} \neq 0 \text{) WRITE (6, CMDMP)}

\text{RETURN}

\text{FORMAT (//I1H ***ICOUNT=+.I5)}

\text{FORMAT (//H0NEGATIVE CONCENTRATION ENCOUNTERED IN COMM)}

\text{FORMAT (//H0SU2 LESS THAN 0 IN COMM)}

\text{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), NNG(100), ISEQ(100), NNKQ, I(50)

M = L - J
J = I(J)
IF (ISIGN(M, L - J), EQ, M) GO TO 10
M = L
L = J
J = M
10 RETURN
END
SUBROUTINE DERIVS
LOGICAL ERR, FAILED
REAL ACOM(30), ELMEN(10), HPI(20)
DOUBLE PRECISION AA, CAPX, CDJ
COMMON AA(22, 24), DJIJ(20, 10), CAPX(20)
COMMON C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CHP
COMMON CTP, CRTP, CTY, CX, CXB, CXMAX, DATEST, DTTEST, DEL1
COMMON DELT2, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT
COMMON PCTEST, PRES, PRESA, PRESB, PREST, PRHC, RHAP, RHUD, ROHB
COMMON QCBAR, RHOC, RHOP, RHP, RHT, RORAR, ROBAR, SCGP, SHT
COMMON SC, SD, SL, SL64, SM, SU, SU2, SUMG
COMMON TEST, TESTB, PRTUNIT, TSTOP, UP, ZP, ZPA
COMMON BE(64), BET(20), BLBK(31), CA(64), CAPXTH(20)
COMMON CPJ(20), CEACT(64), CCI(20), CGM(20), CHI(64), CHIII(20)
COMMON CLNT(64), CLNIT(20), CMW(20), ETAJ(64), ETAJ2(20), GJ(20)
COMMON GJ(20), PERTGJ(20), PGJ(20), PI(64), PCHI(64), QM(20)
COMMON QQ(64), SAI(20), SDCHI(64), SENT(20), SHJ(20), SHJ2(20)
COMMON SKIL(20), SS(20), TB(30), TFA(20), TF8(20), TFC(20)
COMMON TFD(20), TFE(20), TFK(20), THEV(20), XMJAT(20), XNU1(64)
COMMON BE(64), BET(20), BLBK(31), CA(64), CAPXTH(20)
COMMON IC, IM, INEQ, INEOV, IP, IRUN, ISC, ISCP1
COMMON ISMC, ISMCR, ISR, ISS, ISSN, ISSP1, ISSP2, ISSP3, ISSP4
COMMON ISW1, ISW2B, ISW2, ISW2A, ISW3B, ISW3, ISW4A, ISW4B, ISW5A
COMMON ISW5B, ISW6A, ISW6B, IUPD, IZERO, JJK, LC, ML, NFIT
COMMON INI, NNP, NNS, NOS, NOT, NTEST
COMMON IGJ(20), IGJ2(20), IG2(20), KUR(64, 20), LPIJ(20, 10)
COMMON ACOR, ELEMNT, HP
COMMON /BL DELT2(21), BLNT(21), XZER, TWALL, CPWALL, VISROT, DIAM(2)
COMMON SW, RO, JOIM, IPCINT
COMMON ERROR, ERR
COMMON /ERROR/ FAILED
COMMON /AREA/ ATPI(11, 2), PARAM(3, 12, 2), RTHCM(2), NSECT(2)
COMMON /INDEX/ ATPI(11, 2), PARAM(3, 12, 2), NSECT(2)
COMMON INEQ/ ATPI(11, 2), PARAM(3, 12, 2), NSECT(2)
COMMON INEQ/ ATPI(11, 2), PARAM(3, 12, 2), NSECT(2)
COMMON INEQ/ ATPI(11, 2), PARAM(3, 12, 2), NSECT(2)
COMMONT /INDEX/ ATPI(11, 2), PARAM(3, 12, 2), NSECT(2)
COMMON INEQ/ ATPI(11, 2), PARAM(3, 12, 2), NSECT(2)
COMMON HJ(25), ICH(20), IPA(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
COMMON KTR(25), T4(25), T5(25), T6(25), T7(25)
CTD(1)=CT
DO 40 I=1,ISS
XMJATD(I,1)=XMJAT(I)
IF (NT*EQ.1) GO TO 50
SHJ(1)=SHJE
CCPJ(1)=CPJE
SENT(1)=SENTE
CTD(2)=CTE
40   CALL COMM
   IF (ERR.OR.FAILED) RETURN
   CALL EXACT
   IF (ERR) RETURN
   IF (DLCGA*CX*GE-0.01) GO TO 60
   FAILED=TRUE
   RETURN
   IF (ISW3B.EQ.0.OR.IUPC.EQ.1) GO TO 140
   DO 70 K=1,NPRFLS
   DELI2(K)=DELI1(K)
   CALL DLCALL (,FALSE*)
   IF (EMS) RETURN
   IF (ABS(DEL01(K)-DELI2(K))/DELI2(K)EQ.0.01) GO TO 80
   IF (NPRFLS.EQ.1) GO TO 130
   IF (ABS1E-DELI1(K)) LE.0.01) GO TO 130
   70
   DO 90 K=1,NPRFLS
   DEL02(K)=DELI1(K)
   CALL DLCALL (,FALSE*)
   IF (EMS) RETURN
   IF (ABS(DEL02(K)-DELI1(K))/DELI1(K)EQ.0.01) GO TO 120
   IF (NPRFLS.EQ.1) GO TO 110
   DO 100 K=1,NPRFLS
   DEN=DEL02(K)-DELI01(K)-DELI2(K)+DELI1(K)
   IF (DEN*EQ.0) GO TO 120
   100
   DEL01(K)=DELI1(K)+DELI2(K)+DELI01(K)*DELI2(K))/DEN
   GO TO 10
   WRITE (6,150)
   WRITE (6,DRVDMP)
   CONTINUE
   RETURN
   FORMAT (39H BOUNDARY LAYER ITERATION NOT CONVERGED)
END
SUBROUTINE DSMSC (A,I,N)
DOUBLE PRECISION A

C
DATA RNAME /6HDSMSOL /
DIMENSION A(22,24)

DO 20 K=1,I
DO 10 J=1,I
L=L/N
LM=L-N*LN
LN1=LN+1
A(LM,LN1)=A(J,K)

10 IF (K*EO*1) A(J,N1)=A(J,M1)
CONTINUE
CALL DSMQ (A(I,1),A(1,N1),I,KS)
IF (KS*EO*C) GO TO 30
WRITE (6,50)
CALL DUMP (RNAME)
RETURN

30 DO 40 J=1,I
A(J,M1)=A(J,N1)
RETURN

40 FORMAT (36HOMATRIX OF COEFFICIENTS IS SINGULAR.)
END
SUBROUTINE DUMP (RNAME)
LOGICAL ERR
COMMON /ERROR/ ERR
ERR=TRUE.
RITE (6.10) RNAME
RETURN

C
C
C
10 FORMAT (A192)  DUMP ROUTINE CALLED BY "A6"
END

DUMP DUMP DUMP DUMP DUMP DUMP DUMP
1 KTR(25), ICH(20), IPA(25)
2 COMMON /NDCA/ XKJATD(2C.2), CLNTO(2), CTD(2)
3 COMMON /TNERK/ SOTE, CTB, OFMAIX
4 COMMON /INCA/ SUMGH, SCGGH, ODP, ODPE
5 COMMON /INGNE/ AIN(10,10)
6 NAMELST /DMP1/
7 1 A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO
8 2 C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP
9 3 CRB, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMXX
10 4 CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBSTEST, DETL
11 5 DELT, DELTAX, DLOG, DT, ENU, FLUX, HELIX, RPL, R.T.
12 6 PCTEST, PREST, PRES, PRSTH, PRH, RHAP, RHDT, ROB, ROE
13 7 RHDHAR, RHD, RPPL, RBML, RTH, RDBARA, RDBARP, SCPG, SDT
14 8 SEN, SHPG, SC, SL, SL64, SH, SU, SU2, SUMG
15 9 TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA
16 NAMELST /DMP2/
17 1 BET, BLK, CAI, CAPX, CAPXT, CSCPJ
18 2 CEACT, CGI, CGMU, CH, CHI, CLN1MC, CLNFI, C.W, ETAI, ETAJ, GJA
19 3 GJB, PERTG, PGJ, PI, PITCH, IQM, QQ, SAI, SSCI, SENT, SHJ
20 4 SHJA, SKIL, SS, TB, TFA, TFC, TFE, TFK, THT, TPE, TFX, VMAX, XNUI
21 NAMELST /DMP3/
22 1 IC, IN, INFO, INEOV, IP, IRUN, ISC, ISCPJ
23 2 ISMC, ISMCNR, ISR, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4
24 3 ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A
25 4 ISW5U, ISW5A, ISW6A, ISW6B, IUD, IZER0, JAK, LC, M1, M1, MIT
26 5 NIT, NN, NNS, NO, NOT, NTST, 1G, J, J, JM1, K, K1, K, KEEP, M, M, MAP
27 NAMELST /DMP4/ DELBL, BINTL, TWALL, CWALL, VISROT, H0
28 1 DIAM, SW, JDIM, IPOINT, VIS, SAV, XZER0, PPREF, HR
29 2 UVST, DELL, ZCAP, SEN, ZCH, ZCM, ZPZ, ZCH, ZCM, ZPZ, ZCH, ZCM, ZPZ, ZCH
30 3 TFP, EPSF, THETA, SN, XSN, NEP, OWD, TAU, SHJ, DJ
31 4 XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR, XPR
32 5 K0D, PAT, PRAM, RTHCM, NSFCT, ISP, HI, SIGMA
33 6 FLEX, FACP, NMODP, TSDIAM, NT, TSB, NPROF, NPROF, NLN, NORTA, NSUB
34 7 S1, S2, KSA, SUMGH, SCGGH, ODP, ODPE, SETE, CTE, CTE, DTE, BPAR
35 8 EPAR, NT, ITA, KTF, KTR, ICH, IPA, XMJATD, CLND, CTD, OFMAIX, AIN
36 9 FAILED, NORGAC, XI, GONIC, DELBL, DDLBP, DDDLP, AMPP, AMP, COPLD, ISMD
37 WRITE (6, DMP1)
38 WRITE (6, DMP2)
39 WRITE (6, DMP3)
40 WRITE (6, DMP4)
41 IF ( *NOT*, DATAP) GO TO 24
42 DO 10 I=1, NRCOUT
43 BACKSPACE IPTOT
44 WRITE (6, 30) IPTOT, NRCOUT, NRCOUT=0
45 WRITE (6, 40)
46 RETURN
47 CC
48 CC
49 FORMAT (11H0****TAPE, 13, 14H BACKSPACED BY, I*4H RECORDS TO DELE
50 DATA FROM FAILED CASE)
51 END
SUBROUTINE ELCONG (SIGMA, SSIG)
SUBROUTINE ELCONG COMPUTES ELECTRICAL CONDUCTIVITY IN MHOS/CM.

COMMON /TRANS1/ T(3,20,20), ZM2(20)
COMMON /TRANS3/ B(20,3), BR(20,2), AI(2), X(20), DH(20)

SIGMA=0.
IF (IELEC.EQ.0) GO TO 20

DO 10 I=2,N

10 SIGMA=SIGMA+X(I)*Q(1,1,I)

SIGMA=SIGMA*X(1)/SIGMA

RETURN
END
SUBROUTINE E \(VE\) (ET, IP)

TIMING ROUTI.

**FT**=ELAPSED TIME FROM BEGINNING OF EXECUTION IN SECONDS

**IP**=CONTROL FOR PRINTING THE TIME

- **IP** = 0, SUPPRESS PRINT

LOGICAL CALLED
DATA CALLED **/FALSE** /

**ETP**=ETO
**ETO**=ET

IF **CALLED** GO TO 10

**ETO**=0

CALL **RESET**
**CALLED**=**TRUE**

**ET**=0

GO TO 30

10 CALL **TIME** (I)

**ET**=9.001*

IF **IP**=**0** GO TO 20

**ETM**=**ET**/60*

**DET**=**ET**-**ETO**

**WRITE** (6,10) **ETM**, **DET**

GO TO 30

20 **ETO**=**ETP**

30 RETURN

C

40 FORMAT \(24H0************\) ELAPSED TIME=,**F6.2,28H MINUTES SINCE START** \(OETL\) 29

IF \(RUN**, **F5.0,32H SECONDS SINCE LAST PRINTED TIME/\)

END
SUBROUTINE EPART (IR, TE, EF, ER, OF, OR)

COMMON / TNONEO, CTE, DTE, EPAR, EPAR(2, 25), NT, ITR(25), KTF(25),
  ICH(20), IC(20), IFA(25)

DEFINITIONS OF ARGUMENTS

INPUTS
  IR = INDEX OF REACTIONS
  TE = ELECTRON TEMPERATURE (DEG. K)

OUTPUTS
  EF = 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(IR)
GO TO (10, 20, 50, 20, 80, 90, IT)
10 EF=-EPAR(IR)*RO*TE
  GO TO 30
20 EF=-TR*TE
  IF (IT=EQ=4) GO TO 70
30 ER=0*
  QF=EO-EF
40 OR=0*
  GO TO 100
50 ER=0*
  QF=0*
  GO TO 40
60 ER=EF
  QF=0*
  GO TO 70
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 EQUILIBRUM FOR SPECIFIED T AND P.*
T=TEMPERATURE IN DEG. K
P=PRESSURE IN ATM.
REAL ACCM(30), ELEMENT(10), HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22,24), CDIJ(20,10), CAPX(20)
COMMON A, AFN0, AMACH, AR, ARG0, BZERO
1 C, CH, CHA, CHN, CM, CMA, CRA, CRE
2 CRR0, CRR, CSI, CT, CTAP, CTB, CTC, CTMAX, CTMX
3 CTP, CPL, CTX, CX, CMAX, DATE, DATE, DEL1
4 DELT2, DFLAX, DLOGA, DLOGR, DT, EMT, FLUX, MDELX, MCT
5 PCTEST, PRES, PRESH, PRESTH, PRINC, RAPET, RHE, RHO
6 RHOBAR, RHOC, RMP, RMPL, RHX, ROBARI, ROBAP, SCPR, SRT
7 SEN, SMPX, SC, SL, SL64, SM, SU, EU, SUM, USE
8 TEST, TEXIT, TXIT, TSTOP, TXTP
COMMON BE(64), BET(20), BLBK(31), CAI(64), CAPXTH(20)
1 CCPJ(20), CEAI(64), CGL(20), GML(20), CHI(64), CHII(20)
2 CLNMC(64), CLNP1(64), CMX(20), ETAI(64), ETAJ(20), GJ(20)
3 GJ(20), PERTGJ(20), PGJ(20), PI(64), PIH(64), GM(20)
4 GQ(64), SAJ(20), SDCH(64), SENT(20), SHJ(20), SHJAL(20)
5 SKIL(20), SS(20), TBB(30), TFA(20), TFB(20), TFC(20)
6 TFJ(20), TFE(20), TFJ(20), THEV(20), XMJAT(20), XNLJ(64)
COMMON BETP(64,22), EJ(10,20), GFLJ(10,20)
1 XNLJ(64,20), XNLJ(64,20)
COMMON IC, IM, INFO, INFOV, IP, INUN, ISE, ISC, ISCP
1 ISMC, ISCNR, ISE, ISS, IMNR, ISSP, IISP, ISSP3, ISSP4
2 ISN+1, ISN+2, ISWZ, ISWZ, ISW4, ISW4, ISW5, ISW5
3 ISW6, ISW7, ISW8, ISW8, IUPD, IZFD, IZK, LC, M1, MNP
4 NIT, NN, NNS, NOS, NTEST
COMMON IGJ(20), IGJ(20), ITR(5), KUR(64,20), LPIJ(20,10)
COMMON ACOS, ELEMENT, HP
DIMENSION AAA(22,24), BTA(64,20), CAPQ(31), CCI(20), DJ1(20)
1 GDJ(20), SDGJ(20), SHJAP(20), THEV(20)
EQUIVALENCE (AA(1,1), AAI(1,1)), (BETA(1,1), BTA(1,1))
2 (ULBK(1), CAPQ(1)), (CAPXTH(1), CCI(1)), (GJ(1), GJ(1))
3 (CAPX(1), GJ(1)), (SAJ(1), SJ(1)), (SS(1), SDGJ(1))
COMMON /ERROR/ ERR
COMMON /EOC/ ZZCAP(20), ZSEN, ZCH, ZCM, ZRRP, ZRH
COMMON /EOC2/ ZP, ZGML(20)
DATA RNAME, 6 MEGCALC *, CTSAVE *=CT
CT=T/CTAP
DO 10 I=1,10
10 ZCAP(I)=EQM(I)
IF (EQM(I) LE 0.) ZCAP(I)=1.E-3
ZGML(I)=ALG(ZCAP(I))
ZP=ALG(ZP)
CALL THERM
IF (ISMC, EQM) GO TO 60
DO 30 I=1,10
30 ISMC=1
CHI(I)=-XMJAT(IL)+ZP*BE(I)
ECC 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
ZSEN=-ZPZ
ZCM=0.
DO 230 I=1,ISS
ZSEN=ZSEN+ZCAP(I)*(SENT(I)-ZGMU(I))
230 ZCM=ZCM+ZCAP(I)*CGI(I)
ZRBP=P*ZCM/92.058
ZRHO=ZRBP/(1.0+ZRBP*BZERO)
ZCH=ZCH+(ZRBP/ZRHO-1.0)
CT=CTSAVE
RETURN
240 WRITE (6,250)
CALL DUMP (RNAME)
RETURN
250 FORMAT (3SH1 TO0 MANY NEWTON-RAPHSON ITERATIONS)
END
SUBROUTINE EXACT
LOGICAL ERR
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,
1 CARP, CM, CHA, CLNT, CM, CAA, CRA, CRP, EXA
2 CRRH, CRS, CSTA, CT, CTAP, CTD, CT, CTMAX, CTMX, EXA
3 CTP, CTPL, CTT, CX, CXH, CXMAX, DATEST, DBTEST, DELT1,
4 DELTA, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HDELX, PCT,
5 EPCTEST, PERS, PERSA, PERSB, PERTH, PHAQ, RHAP, RHO,
6 RHOMAR, RHOC, RMPL, RTH, RTHOBA, ROBADP, SCPG, SDT,
7 SEN, SHPG, SC, SL, SL64, SM, SU, SU2, SUMG, EXA
8 TEST, TESTB, TPRINT, TSTOP, UP, ZP, ZPA
COMMON BE(64), BET(20), BLBK(31), CAI(64), CAPXTH(20), EXA
1 CLNJ(20), CEACT(64), CGI(20), CGMU(20), CHI(64), CHII(20), EXA
2 CLNJMC(64), CLNPI(64), CMW(20), ETAI(64), ETAJ(20), GJA(20), EXA
3 GJB(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), QJ(: : ), EXA
4 QO(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJ(20), EXA
5 SKIL(20), SS(20), TB(30), TFA(20), TFB(20), TFC(20), EXA
6 TFD(20), TFD(20), TFD(20), THEV(20), XMJAT(20), XNUI(64), EXA
7 COMMON BETA(64, 20), ELJ(10, 20), GELJ(10, 20), EXA
1 XNUIJ(64, 20), XNUIJP(64, 20), EXA
COMMON IC, IM, INEG, INEVG, IP, IRUN, ISC, ISCP1, EXA
1 ISMC, ISMCN, ISR, I5, I5N, ISPA, ISP, ISP3, ISP4, EXA
2 I5S, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, ISW5A, EXA
3 I5S, ISW1A, ISW1B, ISW2A, ISW2B, ISW3A, ISW3B, ISW4A, ISW4B, EXA
4 NIT, NNN, NNS, NOT, NTEST, EXA
COMMON IGL(20), EXA, IGM(20), ITD(5), KUR(64, 20), LPIJ(20, 10), EXA
1 COMMON ACOM, ELEMENT, MP, EXA
DIMENSION AAA(22, 24), BTA(64, 20), CAPQ(31), CCII(20), DGJ(20), GJ(20), EXA
1 И(20), SDGJ(20), SHJAP(20), THEVP(20), EXA
2 EQUIVALENCE (AA(I, J), AAA(I, J), TETE(I, J), ETA(I, J)), EXA
1 (CAP(1, J), GJ1(I), SAJ1(I, J), SB(1, J), SS1(I), SDDG1(J)), EXA
2 (SHJ1(I), SHJAP1(I), THEV(1), THEVP(1)), EXA
COMMON IZIPR, SHJDG, EXA
COMMON YERR, ERRR, EXA
COMMON /SS/ CA5, US, EXA
COMMON /THERK/ SDTE, CTEB, DCHA, CHB, SDCHA, DOMAX, IFAIL, EXA
COMMON /TNNEGQ/ CTE, DTE, BPAR, DPAR(2, 25), NT, ITTR(25), KTF(25), EXA
1 KTR(25), ICH(20), IPA(25), EXA
2 COMMON /TNFO/ SUMG, SPCP, COPR, ODPE, EXA
NAMELIST /EXTMP/ CT, CMA, SU2, DLOGA, DLOGR, SUMG, SHJDO, SUMTDG, EXA
1 IUPD, NT, DTE, DGJ, PICH, SHJ, EXA
? I=SU2/CM4, EXA
3 IF (INEQ KF0) GO TO 100, EXA
4 DO 10 I=1, ISSP2, EXA
5 DO 10 J=1, ISSP4, EXA
6 DO 10 J=1, ISSP4, EXA
7 DO 10 J=1, ISSP4, EXA
8 DO 10 J=1, ISSP4, EXA
9 DO 10 J=1, ISSP4, EXA
10 AAA(I, J)=0, EXA
11 AAA(I, J)=0, EXA
12 DO 20 I=1, ISC, EXA
13 DO 20 J=1, ISS, EXA
14 AAA(I, J)=LPJ(1, J), EXA
15 DO 20 I=ISC1, ISS, EXA
16 Κ=I-ISC, EXA
S1 = BET(K) * SHJ(I) / CT
DO 30 J = 1, ISC
AAA(1, J) = CDIJ(K, J) / GJ(J)
S1 = S1 - CDIJ(K, J) * SHJ(J) / CT
DO 50 J = ISC + 1, ISS
IF (I - J) .GT. 50, 50
AAA(1, J) = -1.0 * GJ(J)
50 CONTINUE
AAA(1, ISS + 1) = S1 / CT
AAA(1, ISS + 2) = BET(K)
CONTINUE
DO 70 J = 1, ISS
AAA(ISSP1, J) = 1.0
AAA(ISSP2, J) = SHJ(J)
AAA(ISSP1, ISSP1) = SUMG / CT
AAA(ISSP1, ISSP2) = SUMG - ZI / CT
AAA(ISSP1, ISSP3) = ZI / CT
AAA(ISSP2, ISSP1) = SCPG
AAA(ISSP2, ISSP2) = ZI
AAA(ISSP2, ISSP3) = -ZI
DO 80 I = ISSP1, ISSP2
80 AAA(1, ISSP2) = DLOGA * AAA(I, ISSP3)
CALL DSMRLO (AAA, ISSP2, 22)
IF (EHR) RETURN
DO 90 I = 1, ISS
90 DGJ(I) = AAA(I, ISSP3)
DT = AAA(ISSP1, ISSP3)
DL3GR = AAA(ISSP2, ISSP3)
GO TO 100
100 NE = NT + 1
IF (NT .EQ. 2) DCHA = -(CMAS / CRS / RHAP / US) * QDPR / RHO / SU
SHJDGJ = 0
SUMTDI = 0
DO 120 J = 1, ISS
DGJ(J) = 0
DO 110 I = 1, ISR
DGJ(J) = DGJ(J) + BETA(I, J) * PICHJ(I)
SHJDGJ = SHJDGJ + SHJ(J) * DGJ(J)
IF (J .LE. NT) GO TO 120
SUMTDG = SUMTDG + DGJ(J)
120 CONTINUE
SUMTDG = CT * SUMTDG
IF (NT .EQ. 2) SUMTDG = SUMTDG + CTE * DGJ(1)
ECDF = SQRT (CMAS / 8.31434E7 / CTAP) / RHP / SU / CRS
AAA(1, 1) = SCPG
IF (NT .EQ. 2) AAA(1, 3) = CCPJ(1) * GJ(1)
AAA(1, 2) = -ZI
IF (IUD0 .EQ. 1) GO TO 130
AAA(1, IF + 1) = ZI * DLOGA * SHJDGJ
IF (NT .EQ. 1) GO TO 130
AAA(1, 4) = AAA(1, 4) - ECDF * QDPR
130 AAA(2, 1) = SUMGH
IF (NT .EQ. 2) AAA(2, 3) = GJ(1)
IF (IUD0 .EQ. 1) GO TO 140
AAA(2, 2) = CT * SUMGH - ZI + (NT - 1) * GJ(1) * CTE
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. IF DOWNSTREAM SOLUTION IS DESIRED
UPDOWN = -1. 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, FRR
DIMENSION Z(2), DZDX(2), RNAME(2)
DATA RNAME /5 + FINDX, 6HFIINDX/
NAMESLIST /DMP/ V, AR, DADX, VO, VL, VU, ERRL, EERRU, 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) .LT. 40, 50
AR = 6.170, A
40
CALL DUMP (RNAME(ENTRY))
50
X = 0.
RETURN
VL = 0.
ERRL = 1 - A
UPPER = .FALSE.
ERRU = 1.0E30
Newton-Raphson I teration TO SOLVE FOR X
60
N = N + 1
IF (N .LE. 50) GO TO 70
WRITE (6, 180) A, UPDOWN, ENTRY, MBL
WRITE (6, DMP)
GO TO 30
70
GO TO (80, 130), ENTRY
80
CALL GEOMAR (SIGN(V, UPDOWN), AR, DADX)
IF (ERR) RETURN
ERR = AR - A
IF (ERR) 90, 160, 110
IF (ERR) 90, 160, 110
100
VL = V
GO TO 140
110
UPPER = .TRUE.
IF (ERR) 120, 140, 140
120
ERRU = ERR
VU = V
GO TO 140
130
CALL GMAR3 (V, DZDX(1), DZDX(2), Z(1), Z(2))
AR = Z(MBL)
DADX=OZDX(MSL)

140 IF (ABS(1.-AR/A)*LE.1.E-5) GO TO 160
    IF (N.LE.10) GO TO 150
    IF (IENTRY.EQ.2.OR..NOT..UPPER) GO TO 150
    V=VL-(VU-VL)*ERRL/(ERRU-ERRL)
    GO TO 60
150  VO=V
    V=V+(A-AR)/ABS(CADX)
    V=MIN(V,2.*VL)
    GO TO 60
160  X=SIGN(V,UPDOWN)
    RETURN

C
170 FORMAT (49HOFINDX CALLED WITH AN AREA RATIO LESS THAN UNITY,,10X,2FDX
    1HA=1PE15.8)
180 FORMAT (33HOMORE THAN 50 ITERATIONS IN FINDX,10X,2HA=,1PD15.8,10X,2FDX
    17HUPDOWN=,0PF3,0,10X,7HIENTRY=,12,10X,4HMML=,12)
    END
COMMON /BL/ DELBL(2),BLINT(2),XZERO,ALL,CPWALL,VISROT,DIAM(2),FRO
1 SW,XO,JDIM,IPCINT
COMMON /SS/ CAS,XUS
COMMON /NC/ CMDIST(2),DDELBL(2)
COMMON /SW,TCH/XPB,DXPB,W
COMMON /AVG/ WSAVE
COMMON /FRO/EDG,ANGLE(10),PADLE(5),WX1,DXW,XW1(20),TANG,WR,
*BLINT(2)*XZERO*
t-.1ALL~CPWALL~VISROT~DIAM(2)
1 SW*WO.
JDIM*IPCINT
CC.'lMZN
CCr4'ON
/SS/
CASeUS
CCr4'ON
/NEO/
GMCST(21
*DDELBL(2)
98
CO1.'tL'ON
/Sc
TCH/
XPP
*DXXE*
W
CCMVON
/AVG/
WSAVE
CC0.'ta40N
/RDhEDG/
ANGLE(10)+PADLE,1
*DXPE*
WX1
*DWX*WXI(20)
*DXP*B
WX1
11WE3GE*bK*
I
N<iX*KANGLF
*XPRLEI
WEDGEM*AXISYF
365X
ISW93
NAMELIST /FROI.P/
CT*ISCNIC.FLAG*PRES~HHO*SU*AMACH*AFNX*XLAST,
1 DELBL,DELBLO,XO,DELBLO,XP,DXVAR,ISTRRT,W
NAMELIST /FROYP2/ XMODEL,XI,X?,CT,CT1,CT2,NUMPI,XFRAC,DELBLO,
1 DELBL,DELBLO,ERRX
C FROZEN FLOW SOLUTION
ENTRY FROZEN
ISOLN=1
C DETERMINE SONIC CONDITIONS IN FROZEN FLOW
DDEL=DELT
F1=0.0
T1=1.0
CT=T1-DELT
S1=0.0
CLNMX=ALOG(CMA)
CO 10 J=1,ISS
SS(J)=ALOG(GJA(J))
10 S1=SI-GJA(J)*(SS(J)+CLNMA)
S2=-(ZPA/CMA)
14T=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-TE1TB) 90,50,50
50 DELT=DELT/2.0
CT=T1-DELT
GO TO 30
60 IF (DELT-TESTB) 90,70,70
70 T1=T2
F1=F2
DELT=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)*CPO0001) GO TO 100
CT=12

142-
GO TO 110
100 CT=T2*DELT*(F3-F1)/(2.0*Z)
110 CONTINUE
CALL THERM
CALL PROP
SM=FLUX
CTMAX=CT
FRESH=PRES
RH(MH)=PRES
IPASS=2
GO TO 120
C EQUILIBRIUM FLOW SOLUTION
ENTRY EQUIL(IPASS)
ISCLN=2
INSTRT=1
#SAVE
130 DO 140 J=1,ISS
140 GJ(J)=GJA(J)
ISYNCE=1
LASTRT=FALSE
SKIP=FALSE
DELTsv=DELT1
IF (ISW3H*NE.0) DELT1=0.049*(1.-CTMAX)
XM01=XMP1
ITS=1
TSTEPC=FALSE
MOLPRT=FALSE
CALL NEXTMP (ITS,XMCD1,XMODEL)
INEC=C
T2E=CTAP
DO 150 I=1,13
150 BLVAR(I)=2
EPSLON=C
T2F=CTAP
EPSF=C
DO 160 I=1, NPRFLS
160 OMIST(I)=1.
IS*TCFL=0
SWITCH=FALSE
IPINT=1
PRES=1.E-0
CT=1.0
RH0=1.0
WH03AP=ROBARA
SU=0.0
AYACM=0.0
AFNA=C.0
RH03=1.0
PRES=1.0
DELTv=DELT1
X3=2.
XP=0.
FLAG=-1.
IF (IPASS.EQ.1) GO TO 180
WRITE (6, 630) (SOLN(N, ISCLN), N = 1, 3) FRO 166
DO 170 J = 1, ISS FRO 167
170  SAVEC(J) = GJA(J) * CMA FRO 168
AMACH = 0. FRO 169
AFNX = 1. * E38 FRO 170
VISC = VISCR FRO 171
SIGMA = SIGR FRO 172
CX = -1. * E10 FRO 173
CM = CMA FRO 174
CALL OUT2 (ISCLN) FRO 175
IF (ERR) RETURN FRO 176
C FRO 177
DECREMENT TEMPERATURE TO GENERATE SOLUTION FRO 178
190 IF (NOT TSTEPC AND (XO-XP) * LT IO,) GO TO 190 FRO 179
TSTEPC = TRUE, FRO 180
DELT = AMAX1(DELTV, IC * DELTV/(XO-XP)) FRO 181
DELT = AMAX1(DELTV, C02 * DELT1) FRO 182
DELT = AMAX1(DELTV / 01 * CT) FRO 183
CT = CT + DELTV FRO 184
IF (ISONIC * EQ -1. AND CT LE CTMAX) GO TO 200 FRO 185
CT = CT + TSTOP) GO TO 610 FRO 186
GO TO 220 FRO 187
C FRO 188
THREAD HAS BEEN REACHED, COMPUTE CONDITIONS AT THROAT FRO 189
200 IF (ISWRB * NF * 0) GO TO 210 FRO 190
FLAG = 1. FRO 191
ISONIC = 1. FRO 192
GO TO 220 FRO 193
210 C FRO 194
CTSTEP = CT + DELTV FRO 195
CTMAX = CTMAX FRO 196
ISONIC = 0. FRO 197
C FRO 198
COMPUTE THE FLOW CONDITIONS FOR TEMPERATURE CT FRO 199
220 GO TO (230, 240), ISCLN FRO 200
230 CALL THRMM FRO 201
CALL PRED FRO 202
GO TO 250 FRO 203
240 CALL NEXTRAP FRO 204
IF (ERR) RETURN FRO 205
AFNX = AMAX1(SM, FLUX, ONE) FRO 206
IF (PRESS <= 0. * PRES) GO TO 260 FRO 207
AMACH = SU * (SORT((FH0 * ABS(ALCG((FH0 / ROB) / ALG(PRES/PRESB)) / PRES) FRO 208
PRESS = PRES FRO 209
SU = SM / AFNX FRO 210
AS = SU * (1. / AMACH) FRO 211
CAS = AS * AS / CT FRO 212
D3 = 290 J = 1, ISS FRO 213
GO TO (270, 280), ISCLN FRO 214
SAVEC(J) = GJA(J) * CMA FRO 215
GO TO 290 FRO 216
270 C FRO 217
DETERMINE X FROM EFFECTIVE AREA RATIO FRO 218
IF (ISWRB * EQU 0) GO TO 310 FRO 219
IF (ISONIC) 310, 320, 300 FRO 220
300 IF (SUM) GO TO 340 FRO 221
IF .CH=ISWHCH+1
IF (ISWHCH.EQ.3) GO TO 330

C  
X BASED ON APPROXIMATING GEOMETRIC AREA RATIO BY AFNX
310 CALL FDNX (AFNX*FLAG*CX)
IF (ERM) RETURN
IF (ISWH3D.EQ.0) GO TO 410
IF (CX*LT*XZERO) GO TO 180
GO TO 370

320 CX=0
DELTI=DELSV
DELT=DELT1
GO TO 370

330 X*TCH=TRUF
XPH=XO
DXH=XO-XP
C  
X CORRECTED FOR DISPLACEMENT THICKNESS
340 CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)
IF (ERM) RETURN
IF (M0DLPT) GO TO 360
IF (CX*GT*XLAST) GO TO 370
WRITE (6,650)
IF (IS<28.4Q.0 OR AR*GT.4*AFNX) GO TO 350
INSTR=INSTR+1
IF (INSTR*GT.3) GO TO 350
W=W/2.
WRITE (6,150) W
GO TO 130

350 WRITE (6,FRDMP)
RETURN

360 CALL TRANSP (CT,CTAP,PRES,PRESA)
GO TO 410

370 XLAST=CX
CALL BLAYER (*TPUE*)
IF (ERM) RETURN
IF (IPASS.EQ.1 .AND. JSONIC.NE.0) GO TO 180
IF (I<11) 410,380,400
380 DO 390 =1,NPAFLS
390 CMDSTY=.DELM(L)
IF (IP/EQ.1) RETURN
730 CALL AGSOLN (AFNX,DELBL,FLAG,AR,CX)
410 IF (M0DLPT) GO TO 500
IF (*NOT*SKIP) GO TO 420
SKIP=FALSE
GO TO 550
420 XPH=XO
XPH=CX
IF (ISWH3B.EQ.0) GO TO 440
DO 430 M=1,NPPFLS
DELRLP(M)=DELRLC(M)
430 DELRAL(E)=DELRL(M)
440 IF (CX*LT*XMAX) GO TO 450
XMODEL=CX
XMODEL=CKX
XMODEL=TPUE
450 IF (CX*GE*XMODEL) GO TO 460
IF (SKIP) GO TO 220

FRO 221  
FRO 222  
FRO 223  
FRO 224  
FRO 225  
FRO 226  
FRO 227  
FRO 228  
FRO 229  
FRO 230  
FRO 231  
FRO 232  
FRO 233  
FRO 234  
FRO 235  
FRO 236  
FRO 237  
FRO 238  
FRO 239  
FRO 240  
FRO 241  
FRO 242  
FRO 243  
FRO 244  
FRO 245  
FRO 246  
FRO 247  
FRO 248  
FRO 249  
FRO 250  
FRO 251  
FRO 252  
FRO 253  
FRO 254  
FRO 255  
FRO 256  
FRO 257  
FRO 258  
FRO 259  
FRO 260  
FRO 261  
FRO 262  
FRO 263  
FRO 264  
FRO 265  
FRO 266  
FRO 267  
FRO 268  
FRO 269  
FRO 270  
FRO 271  
FRO 272  
FRO 273  
FRO 274  
FRO 275
IF (ISONIC,NE.0) GO TO 180
CT=CTSTAR
ISONIC=1
FLAG=1
GO TO 180
610 IF (NPRFLS.EQ.0.1) GO TO 620
AXISY=TRUE
CALL MODEL
AXISY=FALSE
620 RETURN
C C C
C C C
630 FORMAT (1H1,3A4,9HSOLUTION/) 
640 FORMAT (2H0X DECREASED IN FREEZ)
650 FORMAT (1H03H RESTART SOLUTION WITH A LARGER AVERAGING DISTANCE FOR 
1 THE BOUNDARY LAYER CORRELATION PARAMETER, NEW W=9.6/1H1 
660 FORMAT (7H ITERATION TO OBTAIN FREE-STREAM SOLUTION AT MODEL POINT 
IT DID NOT CONVERGE,5X,2H=1PE10,3,5X,7HCM) 
END
SUBROUTINE GEOM
REAL ACCM(30), ELEMENT(10), HP(20)
LOGICAL ERR
DOUBLE PRECISION AA, CAPX, CDIJ, CAPX(20)
COMMON AA(22,24), CDIJ(20,10), CAPX(20)
COMMON GE, CARB, APNTS, APNX, AMACH, AR, ARBA, ARBB, BZERO
1 CAR, CH, CHA, CLNT, CM, CMA, CRA, CRP
2 CRBB, CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMXX
3 CTB, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT
4 DELT2, DELTAX, DLOGA, DLOGR, DT, ENT, FLUX, HUEX, PCT, PCT
5 PCTEST, PRES, PREST, PHO, RHAP, RHO, RHO
6 RHOBAR, RHOC, RHOP, RHP, RHPL, RHTH, RORAR, RORAPP, SCGP, SOT
7 SEN, SHPG, SC, SL, SL64, SM, SU, SU2, SUMG
8 TEST, TSTBD, TPRINT, USTP, UP, ZP, ZPA
COMMON BETA(64,20), BETA(20), BLAK(30), CA(64), CAPXTHI(20), G00
1 CPX(20), CEACT(64), CGI(20), CGMU(20), CHI(64), CHI(20)
2 CNVC(64), CLNF(64), CM(64), ETAI(64), ETA(64), GJAI(64)
3 GJBI(64), PERTCJ(64), PGU(64), PT(64), PCHI(64), QMI(64)
4 GQ(64), SAJ(64), SDCHI(64), SENT(64), SHJ(64), SHJ(26)
5 SKIL(20), SS(64), TDI(64), TFA(20), TFI(20), TFC(64)
6 TFD(20), TFK(20), THEV(20), XMJAT(64), XNUI(64)
7 COMMON BETA(64,20), FLJ(10,20), GELJ(10,20), GELJ(10,20)
1 XNUIJ(64,20), XNUIJP(64,20)
COMMON IC, IM, INEO, INEOV, IP, IRUN, ISC, ISCPI
1 ISCA, ISCH, ISCHPR, ISCHPRR, ISEE, ISNNR, ISSP1, ISSP2, ISSP3, ISSP4, ISSP
2 ISW1, ISW1, ISW2, ISW2, ISW3, ISW3, ISW4, ISW5
3 ISW5, ISW6, ISW6, IUPD, IZERO, JJK, LC
4 NT, NNN, NNS, NOS, NOT, NTEST
COMMON IG(2), IG(2), IGB(5), KUR(64,20), LPIJ(20,20)
COMMON ACOM, ELEMENT, HP
COMMON /BL/DDEL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2)
1 SAP, JDM, IPCINT
COMMON /NEO/ GMCST(2), DDELBL(2)
COMMON /AREA, ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2)
1 AINEG, NSECTU(2), ISHAPE(1,2), NPROFL(2), NPROFL, NBL
COMMON /AGEOM, SORTA, S1, S2
COMMON /ERROR, ERR
DIMENSION DTE(2)
DATA RNAMEF' 176G0M'
CALL GEOMAR (ICX, S1, S2)
IF (EWR) RETURN
IF (NPROFL.EQ.1 AND JDM.EQ.1) SORTA=SORT(S1)
IF (ISW3.EQ.0 CR IUPD.EQ.0) GO TO 10
CALL AESCLN (CX)
10 IF (INEO.EQ.0 OR IUPD.EQ.0) GO TO 180
IF (S1.LE.1.0D0) GO TO 50
ARGZ=2.*C*(S1-1.0D0)A
DELHR0=SORT(ARG)
IF (S1.GT.1.0D1) GO TO 60
IF (S2.GT.20.30) GO TO 60
20 SGN=1.
GO TO 40
30 SGN=-1.
40 RHO=RHTH+SGN*SORT(ARG-SGN)
10)JS*SORT(A/C/(A+2.0))DELHRD*3) GO TO 54
GO TO 130
3
ISJ=ISHAPE(K,1)
IF (ISJ*GT*1) GO TO 7C
RAD(I)=PARAM(1,K,1)+PARAM(2,K,1)*X
GO TO (50,100,100,40), ENTRY
40
DRV(I)=PARAM(2,K,1)
GO TO 100
50
RAT=RAD(I)/RTHCM(I)
IF (JDIY*EQ*0 OR NPNFLS*EQ*2) GO TO 60
S1(I)=RAT*2
S2(I)=2*RAT/RTHCM(I)*PARAM(2,K,1)
GO TO 140
60
S1(I)=RAT
S2(I)=PARAM(2,K,1)/RTHCM(I)
GO TO 100
70
DAB=X-PARAM(2,K,1)
Q=E=RT(PARM(3,K,1)**2-DAB**2)
RAD(I)=PARAM(1,K,1)+SN(ISJ-1)*Q
GO TO (60,100,100,40), ENTRY
80
DRV(I)=S1(ISJ-1)*DAB/Q
IF (ENTRY*EQ*4) GO TO 100
WAT=RAC(I)/RTHCM(I)
IF (JDIY*EQ*0 OR NPNFLS*EQ*2) GO TO 90
S1(I)=RAT*2
S2(I)=2*RAT/RTHCM(I)*DRV(I)
GO TO 10
90
S1(I)=RAT
S2(I)=DRV(I)/RTHCM(I)
100
CONTINUE
IF (NPNFLS*EQ*1) GO TO 130
GO TO (120,130,110,110), ENTRY
110
SOTRSA=SORT(RSA)
Y=RAD(I)*SOTRSA
Z=RAD(2)*SOTRSA
IF (ENTRY*EQ*2) GO TO 180
DYDX=DRV(I)*SOTRSA
DZDX=DRV(2)*SOTRSA
GO TO 180
120
ARATIO=G1(I)*S1(2)*RSA
DERIVA=(S1(I)*S2(2)+S1(2)*S2(I))*RSA
GO TO 150
130
IF (ENTRY*EQ*2) GO TO 160
140
ARATIO=S1(I)*RSA
DERIVA=S2(I)*RSA
150
IF (ARATIO*LT*1 AND ARATIO*GE*0.9999) ARATIO=1
IF (ARATIO*GE*1 AND DERIVA*GE*0.1) GO TO 180
**RITE (6,150) X,ARATIO,DERIVA
CALL DUMP (RNAME)
RETURN
160
Y=RAD(1)
IF (JDIY*EQ*0) GO TO 170
170
Y=SORT(RSA)
GO TO 180
180
RETURN
ERROR IN INPUT DATA FOR NOZZLE GEOMETRY. 4H X = 1PE14.6

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE INGS
LOGICAL ERR ING 1
REAL ACOM(30), ELMNT(10), HP(20) ING 2
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ, AIN ING 3
COMMON AA(22, 24), CDIJ(20, 10), CAPX(20), ING 4
COMMON A, AFNT, AFNX, AMACH, AR, ARBA, ARBB, BZERO, ING 5
1 C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP, ING 6
2 CRS, CSTA, CT, CTAP, CTB, CTC, CTMAX, CTMXK, ING 7
3 CTAP, CTB, CTC, CX, CXI, CXI, CXMAX, DATEST, DATEST, DELTA, ING 8
4 DELTA, DELTAX, DLOG, DLGR, DT, ENT, FLUX, MDELX, PCT, ING 9
5 PCTEST, PESS, PESA, PESB, PEST, PRSN, RHAP, RHAP, RHAB, ING 10
6 ARHBAN, RMC, RMPL, RHTH, ROBARA, ROBARP, SCPG, SDT, ING 11
7 SEN, SMNG, SC, SL, SL64, SM, SU, SU2, SUMG, ING 12
8 TEST, TEST, TPRINT, TSTOP, U, UP, ZP, ZPA ING 13
COMMON BE(64), BFT(20), BLSK(31), CAI(64), CAPXTH(20), ING 14
1 CCPJ(20), CEACT(64), CGI(20), CMU(20), CHI(64), CHI(20), ING 15
2 CKNIC(64), CLNPI(64), CM(20), ETA(64), ETAJ(20), GJAJ(20), ING 16
3 CNI(20), PERTIC(20), PGI(20), PI(64), PICH(64), GMS(20), ING 17
4 GQ(64), SAJ(20), SCAH(64), SENT(20), SCKA(20), SHJA(20), ING 18
5 SKIL(22), SSI(20), TH(20), TFA(20), TFR(20), TFC(20), ING 19
6 TFDM(20), TFM(20), TFK(20), TFKM(20), TNJ(20, 20), ING 20
7 COMMON BETA(64, 20), LLJ(10, 20), GELJ(10, 20), ING 21
8 XNYF 64, 20, XNYVIP(64, 20), ING 22
COMMON IC, IM, INEO, INEOV, IP, IPUN, ISC, ISCP1, ING 23
1 ISMC, ISMCR, ISR, ISS, ISSR, ISSPR, ISSPR2, ISSPR3, ISSPR4, ING 24
2 ISKA, ISKA1, ISKAA, ISH, ISKAA, IUPD, 1ZTP, LC, M1, NF1, ING 25
3 IN, NKN, NNS, NQ, NT, NTES, ING 26
COMMON IGI(20), IGM(20), ITB(5), KURT(64, 20), LPIJ(20, 10), ING 27
COMMON ACOM, ELMNT, H, ING 28
DIMENSION AAA(22, 24), BTA(64, 20), CAPO(31), CCI(20), DGI(20), GJ(20), SRF ING 29
1 J(20), SDGJ(20), SHJAP(20), THEVP(20), ING 30
2 EQUIVALENCE (AA, 1, 1, AAA, 1, 1), (BTA, 1, 1), BTA(1, 1), ING 31
3 (BLSK, 1, 1, CAPX(1), (CAPXTH(1), (LIJ(1), 1), GJ(1), 1, DGI(1), 1), ING 32
4 BETA(1, 1), BTA(1, 1), (BJP(1), (CPX(1), (IJA(1), (SP(1), (DGI(1), 1), ING 33
5 (SHJA(1), SHJAP(1), THEVP(1), THEVP(1), ING 34
6 COMMON /EQC/ ZCAP(20), ZSEN, ZCI, ZCM, ZRBP, ZRH ING 35
7 COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 36
8 COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 37
9 COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 38
10 COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 39
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 40
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 41
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 42
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 43
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 44
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 45
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 46
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 47
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 48
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 49
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 50
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 51
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 52
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 53
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 54
COMMON /ZC/ ZCAP(20), ZSEN, ZCS, ZCM, ZRBP, ZRH ING 55
BET(I)=-1.0
DO 40 K=1,ISC
40 BET(I)=BET(I)+CDIJ(I,K)
50 CONTINUE
60 SUM=0.0
DO 80 I=1,ISC
70 QM(I)=CM(I)+AIN(K,I)*CAPQ(K)
SUM=SUM+QM(I)
80 CONTINUE
DO 90 I=1,ISC
90 QM(I)=QM(I)/SUM
CCI(I)=0.0
DO 100 J=1,ISC
100 CGI(I)=CCI(I)
IF (ISMCEQ.0) GO TO 130
DO 120 I=1,ISMC
110 SUM=SUM+CDIJ(I,K)*CGI(K)
120 CONTINUE
130 RETURN
ENTRY INTA
CALL EUCALC (CTAP,PFESA)
IF (ERR) RETURN
ZPA=ZPZ
CM=ZCM
SEN=ZSEN
CHA=ZCH
CMA=CM
CRRH=SEN/CMA
SEN=CRRH*CPA
CHA=CMA
DO 140 I=1,ISS
140 CGI(I)=ZGMU(I)
150 QM(I)=ZCAP(I)
160 CGI(I)=QJAL(I)+CAPX(I)/CMA
RETURN
END
SUBROUTINE INIT
REAL ACXM(30), ELMNT(10), HP(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CIJ
COMMON AA(22,24), CIJ(22,1), CAPX(20)
COMMON A, AFNTS, AFN, AMACH, AR, ARBA, ARBB, BZERO.
1 C, CARB, CH, CHA, CLNT, CM, CMA, CRA, CRP
2 CRH, CRS, CSTA, CT, CTAP, CTB, CTC, CMAX, CTMXX
3 CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELTI
4 DELT, DELTAX, CLGDA, CLGDB, DT, EMT, FLUX, HOELX, PCT
5 FTPTEST, PTES, PRFA, PRE, PREST, PRHC, RHAP, RHO, RHOB
6 RHCBAR, RHDC, RHDP, RHPL, RPTH, ROBAR, ROBAPP, SCPG, SDT
7 SEN, SHF, SC, SL, SL48, SP, SU, SUMG
8 T, TEST, TST, TSTEM, TSTIP, TSTOP, UP, ?PA
9 TST, TEST, TSTEM, TSTOP, UP, ?PA

COMMON PFF(64), BFT(20), BLKX(31), CAT(64), CAPXTH(20)
1 CCPFJ(20), CEACT(44), CGL(20), CGMA(20), CHI(64), CHIT(20)
2 CLNVC(64), CLAPI(64), CMX(20), ETA(64), FTJ(20), GTA(20)
3 GJ(20), PERTGJ(20), PGJ(20), PJ(64), PICH(64), GUM(20)
4 GJ(64), SAJ(22), SCCHI(64), SENT(20), SHJ(20), SHJA(20)
5 SKIL(20), SS(22), TH(30), TFA(20), TFG(20), TFC(20)
6 TFG(20), TFE(20), TFG(20), THEV(20), XMJA(20), XNU(64)
7 COMMON RATA(64,20), FLJ(10,20), GELJ(10,20)
8 XNU(64,20), XNU(64,20)
9 COMMON IC, IM, IEQ, INEOV, IP, IRUN, ISC, ISCP
1 COMMON ISNC, ISNCR, ISH, ISS, ISSP1, ISSP2, ISSP3, ISSP4
2 COMMON ISX, ISX1, ISX2, ISX3, ISX4, ISX5, ISX6, ISX7
3 COMMON ISWA, ISW, IUPC, IZER0, JK, LC, M, NFIT
4 COMMON NN, NNN, NPSNT, NPSNP, NPSNP, NNP
5 COMMON IGJ(20), IGK(20), ITH(5), KUR(64,20), LPIJ(20,10)
6 COMMON ACY, ACY, ACY, ACY
7 DIMENSION AA(22,24), RATA(64,20), CAPQ(31), CCI(20), DGJ(20), GJ(20), SB
8 DIMENSION JI(2,1), SDGJ(2,1), EHJAP(20), THEV(20)
9 EQUIVALENCE (AA(1,1), AA(1,1), (BETA(1,1), BTA(1,1))
1 COMMON (BLK(1), CAPQ(1), (CAPXTH(1), CCI(1)), (GJA(1), DGJ(1))
2 COMMON (CAPX(1), GJ(1)), (SAJ(1), SB(1)), (SSP(1), SDGJ(1))
3 COMMON (SAJ(1), SB(1)), (SHJA(1), SHJA(1))
4 COMMON (THA(1), SHJA(1)), (SHJA(1), SHJA(1))
5 DIMENSION MM(4,3), MM(4,3)
6 COMMON SM, SM, SM, SM
7 COMMON SMAC, SMAC, SMAC, SMAC
8 COMMON MM(4,3), MM(4,3), MM(4,3), MM(4,3)
1 COMMON DNSER, DNSER, DNSER, DNSER
2 COMMON DNSER, DNSER, DNSER, DNSER
3 COMMON DNSER, DNSER, DNSER, DNSER
4 COMMON DNSER, DNSER, DNSER, DNSER
5 COMMON DNSER, DNSER, DNSER, DNSER
6 COMMON DNSER, DNSER, DNSER, DNSER
7 COMMON DNSER, DNSER, DNSER, DNSER
8 COMMON DNSER, DNSER, DNSER, DNSER
9 COMMON DNSER, DNSER, DNSER, DNSER
1 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
2 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
3 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
4 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
5 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
6 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
7 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
8 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
9 COMMON (TAD(1), TAD(1), TAD(1), TAD(1))
TEST=000001
TESTB=000001
CT=1*0
CTE=1*0
JK=0
NQT=0
INE=0
NTES=100
IM=1
CRA=1*9872
CRP=CTA
CTP=ALC(GTAP)
IF (NFT) 10, 30, 10
T1=CTAP
T2=T1*T1
T3=T1*T2
T4=T1*T3
DO 20 I=1*ISS
IF (IGJ(I), EQ, 0) GO TO 20
I=15(I)
TFL(I)=T1*SRP(10, II)
TFL(I)=T2*SRP(11, II)
TFL(I)=T3*SRP(12, II)
TFL(I)=T4*SRP(13, II)
20 CONTINUE
DO 60 I=1*ISS
IF (ETAJ(I), LE, 0) GO TO 60
II=IS(I)
S4(I)=SRP(17, II)+(1+5*ETAJ(I))CTAP
TBM(I)=SPR(18, II)/CTAP
THEV(I)=THEV(I)
DO 40 J=1, 3
THEV(J+1, I)=SRP(J+40, II)/CTAP
K=16(I)
DO 50 L=1, K
50 CONTINUE
DO 70 L=1, ISS
II=IS(I)
S3(J+1)=SRP(15, II)/CRP
ISMC=ISMC
ISSNM=ISS
RETURN
END
SUBROUTINE KANDMU (ZK)

SUBROUTINE KANDMU COMPUTES VISCOSITY AND TRANSLATIONAL COMPONENT OF THERMAL CONDUCTIVITY. INDEX L=1 INDICATES TRANSLATIONAL THERMAL CONDUCTIVITY, L=2 INDICATES VISCOSITY.

DIMENSION ZK(2)

COMMON /TRANS1/ T, O(3, 20, 2C), ZM2(20)

COMMON /TRANS2/ B(20, 2), BR(20, 2), A(2), X(20), DH(20)

COMMON /TRANS3/ N, ILEc, ID1, ID2

C COMPUTE DIAGONAL MATRIX ELEMENTS CAP-A(I, N)

DO 20 I=1, N

30 B(I, I)=C(I, I)

DO 10 J=1, N

10 B(I, J)=B(I, J)*O(3, I, J)

20 B(I, J)=B(I, J)*ZM2(1)

C COMPUTE AVERAGE OF NON-DIAGONAL MATRIX ELEMENTS SMALL-A(I, J)

DO 60 L=1, 2

U=N

V=N

DO 40 J=1, N

40 U=U+X(I, J)*W

V=W+X(I, J)**2

GO TO 50

60 CONTINUE

70 A(I, I)=4*A(I)

C COMPUTE VISCOSITY AND TRANSLATIONAL THERMAL CONDUCTIVITY

DO 90 I=1, 2

80 Y=Y*X(I)/(A(I)+B(I, I))

90 Z=(L)=3, 125*Y/(1-A(I))*Y

RETURN

END
SUBROUTINE KINT (ZKINT,T)
COMMON /TRANS1/ T*(3,20,20),ZM2(20)
COMMON /TRANS3/ B(20,2),BR(20,2),A(2)*X(20),DH(20)
COMMON /TRANS5/ N,IELEC,ID1,ID2
C COMPUTE SUM OVER J OF X(J)*DELTA(I,J)
IF (N*E0<1) GO TO 20
N1=N+1
DO 10 I=2,N
I1=I-1
NI=N-I1
DO 10 J=1,II
NJ=N1-J
DO 10 10 L=1,II
B(I,L)=B(I,L)+X(I)*G(L,J,I)
10 CONTINUE
B(N1,L)=B(N1,L)+X(NJ)*G(L,N1,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,I)
RETURN
END
DJ 140 I=1,ISR
WRITE (6,570) I,KTF(I),KTR(I),ITR(I),EPAR(1,I),EPAR(2,I),IPA(I)
IF (NIT) 160,190,160
WRITE (6,510)
DJ 180 J=1,ISS
IF (IG(J)) 170,180,170
WRITE (6,520) HP(J),TFA(J),TFB(J),TFC(J),TFD(J),TFE(J),TFK(J),SHJAL
IP(J)
190 CONTINUE
190 IF (INEQV.NE.0) WRITE (6,580)
WRITE (6,590) HP(JATCM),HP(JJMJL),(COM1(I,IAMBIP),I=1,2),(COM2(K,LEL
191(I+1,K=1,2)
IF (1@3H.NE.0) GO TO 200
WRITE (6,530)
GO TO 210
WRITE (6,540)
WRITE (6,600)
CXMAX1=CXMAX/2.54
IF (XMODPI.LT.9.5E19) WRITE (6,630) XMODPT,XMODPI,CXMAX1
IF (NTS.EQ.0) GO TO 240
GO TO (220+230), NPFFLS
220 WRITE (6,610) (TSCIAM(I),I=1,NTS)
GO TO 240
WRITE (6,620) (TSDIAM(I),I=1,NTS)
240 IF (.T. T AXISYM) GO TO 250
WRITE (6,640) TMODEL, TPLATE
IF (FSLAG) 250,260,270
WRITE (6,650)
GO TO 280
WRITE (6,660)
GO TO 280
WRITE (6,670)
WRITE (6,680) CATFAC
KDP1=KDTM+1
WRITE (6,690) (WORD(I,KDP1),I=1,3)
290 IF (.NOT. WEDGEEM) GO TO 340
WRITE (6,700) (ANGLE(I),I=1,NANGLE)
WRITE (6,710) (RADLE(I),I=1,NRAOLE)
IF (NWX.EQ.0) GO TO 300
WRITE (6,720) NWX,DWX,DX1
300 IF (WX(I).GT.1.E19) GO TO 330
DO 310 I=1,26
IF (WX(I).GT.1.E19) GO TO 320
310 CONTINUE
320 WRITE (6,730) (WX(I),I=1,NWX)
330 WRITE (6,740) TWEDGE,WK
340 RETURN
C
C
350 FORMAT (1X,H99H ELEMENT
1 ATOM FRACTION ELEMENT MOLECULAR WEIGHTS/
360 FORMAT (1X,31X,A6,15X,1PE14.6,14X,E14.6)
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 /6MHMATINV/
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 = M) 30, 30, 60
30 DO 50 I = K1, M
50 IF (BIG < ABS(A(I, JJ))) 40, 50, 50
40 K = 1
BIG = ABS(A(I, JJ))
50 CONTINUE
60 BIG = A(K, JJ)
IF (BIG < 90.70) 70, CONTINUE
WRITE (6, 300)
CALL DUMP (RNAME)
RETURN
80 IF (K = JJ) 90, 110, 90
90 J = IV(K)
IV(K) = IV(JJ)
IV(JJ) = J
DO 100 J = 1, M
B = A(JJ, J)
A(JJ, JJ) = A(K, JJ)
100 A(K, J) = R
110 A(JJ, JJ) = 1.0
JA = JJ + 1
120 DO 122 J = 1, M
122 A(JJ, JJ) = A(JJ, J) / BIG
JB = J + 1
IF (JB) 170, 170, 130
130 DO 160 I = 1, JB
BB = A(I, JJ)
A(I, JJ) = 0.0
160 A(I, JJ) = BB - B
170 IF (JA = M) 180, 180, 220
180 DO 210 I = JA, M
BB = A(I, JJ)
A(I, JJ) = 0.0
210 CONTINUE
IF (BB) 190, 210, 190
DO 200 J=1, M
200 A(I,J)=A(I,J)-BB*A(JJ,J)
CONTINUE
GO TO 20
M1=M-1
IF (M1) 290, 290, 230
DO 280 J=1, M1
280 A(I,J,M)=0
IF (IV(I)-J) 250, 240, 250
K=1
GO TO 260
CONTINUE
WRITE (6, 310)
CALL DUMP (RNAME)
RETURN
IV(K)=IV(J)
DO 270 L=1, M
AA=A(L,J)
A(L,J)=A(L,K)
270 A(L,K)=AA
CONTINUE
RETURN
FORMAT (1H1, 26H MATINV, MATRIX SINGULAR /)
FORMAT (1H1, 37H INDEXING OR STORAGE FAILURE IN MATINV/)
COMMON /RDMSYC, LEWIS, IAMBIP
COMMON /C0LDM, CGMW, OPJ(10), IJCS(10), NCS
COMMON /RDMSY, HS, SUPGC, MFPTR, NOTRAN
DIMENSION QUANT(20,2)
DATA QUANT /0, T2E, 4HPT2E, 5MUT2E, 3HLEE, 4HST2E, 5MMWT2E,
1 6HS1G2E, 3HPRE, 5HEPSLE, 4HQFE, 4HOFF, 5HOESRI, 6MDLRREF,
2 6MDLRH+4HQFE, 6HOEFH, 5KB2PF, 6MHRT, 2HTSF, 4HPT2F,
3 4HQFFE, 4HOPPF, 5HOFSR, 6MDLREF, 6MDLRFH, 4HOFHE, 4HOFHF,
5 5KH2PF, 5MMHTF, 1H
DIMENSION TITLE2(3,2)
DATA TITLE2 /SHEQUIL, 6HBRUM, 6HSCHO, 0, FROZEN, 6H SHOCK, 1H
DATA EQUVAL /2H =
DATA CATM /1, 01, 125N6/
DATA SOQF /C, 74.7, 6
NAMELIST /MOMDMP/P2E, EPSLON, T2F, EPSF, PS, 2CH, ZCM, ZRMQ, HS, R1, P1,
1 1GOUT, LO, HI, ERZ, ERRLO, ERRRI, TLO, THI, T2, EPS, EPSOLD, H2,
2 ALPHAT, IS, DIS
DIMENSION SCOUT(20)
IF (NOT (AXISYM, OR, WEDGEM)) GO TO 480
IF (A4MDEN, GE, 1, 5) GO TO 10
WRITE (6, 490)
GO TO 480
CONTINUE
NORMAL SHOCK SOLUTION
CALL ELTIME (ET0, 0)
U1=SORTRH(3143607*CTAP, CMA)*SU
R1=RHAPRHO
HD=0, 5*SU
HS=HD+CH
UM=U12*R1+U1**2/CATM
P1=PRES*PRES
PT=P1+R112
PC=4, *R112
T1=CTAP*C
CTSAVE=CT
C41=8, *CSS*R112/P1
IF (EPSLON, GT, 0, 1) GO TO 40
EPSLON=0, 01
IF (NOTRAN) GO TO 40
DO 30 I=1, 1SS
DO 20 K=1, NCS
IF (IJCS(K), NE, 1) GO TO 20
SAVEC(I) = C(I, K)
GO TO 30
CONTINUE
SAVEC(I) = 0
CONTINUE
CALL TRANSX (TMODEL, PFILE)
PRW=PR
WRITE (6, 500) PRW
IF (EPSF, EQ, 0, 0) EPSF=0, 01
DO 440 ISOLN=1, 2
IF (ISOLN, EQ, 1, AND, NOT, AXISYM) GO TO 440
IF (FSTAG, LT, 0, AND, ISOLN, EQ, 1) GO TO 440
CONTINUE
GO TO 440
}
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 EOCALL (T2,P2)
IF (FYR) RETURN
EPSOLD=EPS
EPS=R1/7RHO
IF (ABS(EPS/EPSCLD-1.)*LE.0.001) GO TO 130
ALPHA=P2/ZRHO
DISC=PT**2-PC*ALPHA
IF (DISC.GE.0.0) GO TO 100
EPS=EPSOLD
IF (LJ) GO TO 90
T2=0.9*T2
GO TO 50
90 T2=2.5*(T2+TLO)
GO TO 50
100 P2=P2*SQRT(DISC)
GO TO 90
110 ICOUNT=ICOUNT+1
CT=T2/CTAP
IF (ICOUNT.GT.30) GO TO 450
CALL THERM
ZCM=CM1
ZCM=0.
DO 120 I=1,ISS
ZCM=ZCM+SAVEC(I)*SHJ(I)
120 OP=P1*T2/T1
DISC=PT**2-4.*R1U12*OP
IF (DISC.LT.0.) GO TO 450
P2=0.5*(PT+SQRT(DISC))
R2=R1=P2/OP
EPS=R1/R2
130 H2=HS-HD*EPS**2
ZCM=ZCM*CMA/ZCM
IF (ABS(ZCM/H2-1.)*LE.0.001) GO TO 170
E2=ZCM=H2
140 IF (ERZ) 140,140,150
E2=RC=FRZ
TLO=T2
LD=*TRUE*
IF (HI) GO TO 160
T2=1.1*T2
GO TO 50
150 ERR=ERZ
TH1=T2
HI=TRUE*
IF (LO) GO TO 160
T2=0.9*T2
GO TO 50
160:
T2=T2-ERZ*(THI-TLO)/(ERRHI-ERRLO)
GO TO 50
C
SONIC FLOW FROM THE SHOCK TO T': T STAGNATION POINT ON THE MODEL
PS=PS2+C*S*R1U12*EPS
T5=T2
HI=FALSE*
ICOUNT=0
ICOUNT=ICOUNT+1
IF (ICOUNT.GT.20) GO TO 450
GO TO (190,230)
ISOLN
CALL EQCALC (TS,PS)
IF (ERR) RETURN
GO TO 220
CT=TS/CTAP
CALL THERM
ZCH=0
DO 210 I=1,ISS
ZCH=ZCH+SAVEC(I)*SHJ(I)
210:
ZCH=ZCH+CM*ZCM
IF (ABS(ZCH/HS-1).LE.0.001) GO TO 260
ERZ=ZCH/HS
IF (ERZ) 230,230,240
230:
TLO=TS
ERRC=ERZ
IF (HI) GO TO 250
TS=1.1*TS
GO TO 190
240:
THI=TS
ERRHI=ERZ
HI=TRUE*
250:
TS=TS-ERZ*(THI-TLO)/(ERRHI-ERRLO)
GO TO 180
260:
IF (ISOLN.EQ.0) GO TO 280
IF (.NOT.AXISYM.OR.FSTAG.EQ.0) GO TO 300
ZSEN=4*LOG(PS)
DO 270 I=1,ISS
IF (SAVEC(I).LE.0) GO TO 270
ZSEN=ZSEN+SAVEC(I)*(SENT(I)-ALOG(SAVEC(I)))
270:
CONTINUE
ZPH=S2*PS/P2*T2/TS
C
VGP(1)=V*LOCITY GRADIENT PARAMETER FOR HEMISPHERE
C
VGP(1)=VELOCITY GRADIENT PARAMETER FOR FLAT-FACED MODEL
C
SUDD(1)=STANDOFF DISTANCE FOR HEMISPHERE (CYLINDER IF KDIM=0)
C
SUDD(2)=STANDOFF DISTANCE FOR FLAT-FACED MODEL
VGP(1)=SRT(2)*PS*CATM*(1-PI/PS)/ZRP**0.5
SRT=SORT(FPS)
IF (KDMM.EQ.0) GO TO 290
SUDD(1)=SUDD(1)*(1+3.5*(EPS-0.19)**1.6)
SUDD(2)=1.12*EPS**C.39
290 GO TO 300
291 SOUR(1)=1.0/(2.*EPS/((1.e+0.5)*EPS)*ALOG(1.e333333/EPS)-0.5*EPS)-1.e
292 SOUR(2)=-1.23*SOTE*ALOG(0.79*SOTE)
293 MOD 221
294 VGP(2)=G.64*EPS
295 MOD 223
300 SHV(I,1,ISLN)=E2
301 MOD 224
SHV(2,1,ISLN)=EPS
302 MOD 225
IF (*NOT,AIXSYM* OR (ISCLN,EQ.2*AND,FSTAG,EQ.0)),1) GO TO 440
303 MOD 226
IF (ISCLN,EQ.2) GO TO 320
304 MOD 227
DO 310 I=1,NSS
305 MOD 228
310 SAVEC(I)=ZCAP(I)
306 MOD 229
HCF=0.
307 MOD 230
IF (NOTAN) GO TO 360
308 MOD 231
DO 340 I=1,NSS
309 MOD 232
NEL=0
310 MOD 233
DO 330 J=1,ISC
311 MOD 234
IF (LPIJ(I,J),EQ.J) GO TO 330
312 MOD 235
IF (LPIJ(I,J),NE.1) GO TO 340
313 MOD 236
NEL=NEL+1
314 MOD 237
IF (NEL,GTE.1) GO TO 340
315 MOD 238
CONTINUE
316 MOD 239
HCF=HCF*SAVEC(I)*SHJA(I)
317 MOD 240
340 CONTINUE
318 MOD 241
HCF=HCF*CRP/ZCM
319 MOD 242
CALL TRANS (TS,PS)
320 MOD 243
IF (ERR) RETURN
321 MOD 244
HSTAG=CRP/CMA*2CH
322 MOD 245
H=CPWALL*TMDDL
323 MOD 246
HRIATO=HCF/(HSTAG-HW)
324 MOD 247
IF (LEWIS.EQ.2) HRIATIO=0.
325 MOD 248
qsri=0.75*5OR/PS/(HSTAG-HW)
326 MOD 249
C 3.375=1.8*0.2417, WHERE FACTOR 1.8 CONVERTS ENTHALPIES FROM CAL/GM
327 MOD 250
C TO BTU/LB
328 MOD 251
350 DO 350 L=1,2
329 MOD 252
QSF=0.51/(L-PF#PF)**0.3*SQ2F(KDIM+1)*SQRTR(ZRHO*VISC*U1*VGP(L))*(HSTAMCD
330 MOD 253
1G-HW)
331 MOD 254
QFRE(L)=QS*(1.0+FLEWIS**ELN(L))**HRATIO
332 MOD 255
QFRE(L)=QS*(1.0+FLEWIS**ELN(L))**CATFAC-1.0**HRATIO
333 MOD 256
350 CFSI=EPS/1.4**2*RI*U1/VISC*30.98
351 MOD 257
360 MOD 258
360 SCOUT(I)=TS
361 MOD 259
SCOUT(2)=CF(1)*ZRHO/RHAP
362 MOD 260
SCOUT(3)=PS
363 MOD 261
SCOUT(4)=VISC*CF(3)
364 MOD 262
SCOUT(5)=FLEWIS
365 MOD 263
SCOUT(6)=Z^FNCRA/ZCM
366 MOD 264
SCOUT(7)=ZCM
367 MOD 265
SCOUT(8)=SIGMA
368 MOD 266
SCOUT(9)=PR
369 MOD 267
SCOUT(10)=EPS
370 MOD 268
IF (NOTAN) GO TO 370
370 MOD 269
SCOUT(11)=QFRE(2)
370 MOD 270
SCOUT(12)=QFRE(2)
370 MOD 271
370 SCOUT(13)=QSR
370 MOD 272
SCOUT(14)=SOUR(2)
370 MOD 273
SCOUT(15)=SOUR(1)
370 MOD 274
RHAN 1.54)

1 FORMAT (46X,19HMODEL CONDITIONS = .3A6)
111 TEMPERATURE ON MODEL =.F6x4)
MOD 330
510 FORMAT (46X,19HMODEL CONDITIONS = .3A6)
MOD 331
520 FORMAT (46X,19HMODEL CONDITIONS = .3A6)
MOD 332
530 FORMAT (46X,17HMODEL FRACTIONS = .3A6)
MOD 333
540 FORMAT (5(1X,A6,3X,1H=,4X,1PE10,3,3X))
MOD 334
550 FORMAT (5(1X,A6,3X,1H=,4X,1PE10,3,3X))
MOD 335
560 FORMAT (46HFILE DEPENDED TIME IN MODEL AND WEDGE CALCULATIONS =,F4x0,8H)
MOD 336
END

PRINTOUT PAGE IS
OF POOR QUALITY
CALL DSMOSL (AA,M1,22)
IF (ERR) RETURN
NIT=NIT+1
DO 340 K=1,M,ISC
ZB=1.0+AA(K,M2)
IF (ZB) 280, 280, 310
CAPX(K)=CAPX(K)/2
IF (CAPX(K)) 290, 300, 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
CGMU(K)=ALOG(CAPX(K))
GO TO 340
330 WRITE (6,480) K
CGMU(K)=CGMU(K)+ALOG(ZB)
CONTINUE
ZC=1.0+AA(M1,M2)
IF (ZC) 350, 350, 360
PRES=PRES/2
IF (PRES) 370, 380, 370
PRES=PRES*ZC
IF (PRES) 370, 390, 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 (NODE$) GO TO 430
DO 420 I=1,ISMCR
SKIL(I)=CHII(I)*BET(I)*(ZP+ZPA)
DO 410 L=1,ISC
SKIL(I)=SKIL(I)+CDIJ(I,L)*CGMU(L)
J=1+ISC
CGMU(J)=SKIL(I)
SKIL(I)=EXP(SKIL(I))
CAPX(J)=SKIL(I)
CONTINUE
420 DO 450 K=1,M,1
430 IF (ABS(AA(K,M2))$LE.TEST) GO TO 450
IF (NIT$TEST) 70, 70, 440
440 WRITE (6,500)
CALL DUMP (RNAME)
RETURN
450 CONTINUE
CM=0.0
DO 460 I=1,ISSNR
CM=CM+CAPX(I)*CG(I)
460 CM=CH+CAPX(I)*SH(I)
RETURN
RHOBAR = PRES*CMA/(CMA*CT)
RHO = (RHOBAR*RUBARA)/(1.0+RHOBAR*ROBARP*BZERO)
CH = CH*(CMA/C M)
CH = CH*(PKFS/RHOBAR)*(RHOBAR*(ROBARA/RHO)-1.0)
DG 470  I0=IM*ISSNR
           GJ(I)=CAPX(I)/CM
SU2=2.0*(CHA-CH)
SU=SQRT(SU2)
FLUX=RHO*SU
RETURN

C C

480  FORMAT (15H IN NEWRAP,CAPX(*12,3H)=0)
490  FORMAT (45H 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/ ATPI(11,2), PARAM(3,12,2), RTHCM(2), NSECT(2),
NLECTU(2),ISHAPE(12,2), NPROFL(2), NPRFLS, NBL
COMMON /NEWMP/ FACMP, NMODPT
COMMON /MODPT/ TSOIAM(20), TSAR(20), NTS, MBL
COMMON /MODPAR/ XMP1, DXMP, FSTAG, CATFAC, TMODEL, XMODP1, XMDDP,
TPLATE, KDIM

DATA ONE /1e/

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 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
RETURN

END
COMMON /GLIM2/ NOREAC(64)
COMMON /AREA/ ATPI(11, 2), PARAM(3, 12, 2), RTHCM(2), NSECT(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, EPSLON, T2F, EPSF
COMMON /MODP2/ MODLPT
COMMON /THRT/ RSA
COMMON /SWITCH/ XPB, DXPB, W
COMMON /TEMPRY/ SAVEC(20)
COMMON /AVG/ WSAVE
COMMON /STAG/ T2E, EPSLON, T2F, EPSF
COMMON /MODP2/ MODLPT
COMMON /THRT/ RSA
COMMON /SWITCH/ XPB, DXPB, W
COMMON /TEMPRY/ SAVEC(20)
COMMON /AVG/ WSAVE
COMMON /RDWEDELG/ ANGLE(10), RADLE(5), WX1, DWX, WX1(20), TWEDGE, WK,
1 NWX, NANGLE, RADLE, WEDGEM, XAXISYM, ISWSYM
COMMON /TNGNFO/ CTE, DTE, BPAR, EPAR(2, 25), NT, JTR(25), KTF(25),
1 KTR(25), 1CH(20), IPA(25)
COMMON /TNERK/ SOTF, CTEB, DCHA, CHB, SDCHA, DQMAX, IFAIL
COMMON /STEPS/ NSTEPS
COMMON /NEQ/ SUMGH, SCPGH, QDPR, QDPE
COMMON /AEQEM/ SQRTA, S1, S2
COMMON /SS/ CAS, US
DIMENSION DELS(2), IF(30)
COMMON /SS/ CAS, US
DIMENSION DELS(2), IF(30)
DATA TENTH /10000, 1, ONE /1, DO/
WRITE (6, 1C80)
C INITIALIZATIONS
w = WSAVE
QDMAX = 0
QEPOLC = 0
QDPE = 0
NSTEPS = 0
ISTEPS = 0
ISUBD = 0
T2E = CTAP
EPSLON = 0
T2F = CTAP
EPSF = 0
MODLPT = FALSE
XMSET = FALSE
XMODEL = 1.05
IRA = 0
ALOGT = ALOG(TENTH)
ICOUNT = 0
DO 12 I = 1, ISR
 10 NOREAC(I) = FALSE
 1 IPOINT = 1
 2 ISS1 = NWSYM
DELT = DELT1
IF (ISWSYM = 0) DELT1 = AMAX1(0.049*(1.0-CTMAX), DELT1)
HDELX = 0.5*DELTAX
DQLD = DELTAX
DQ 20 I = 1, NPRFLS
20 DDELX (1) = 0
C = 1.0
C = 1.0
SC = 1.
SCD=2
ITB(I)=0
NNS=0
NNN=0
ISC=ISC+1
ISSP2=ISS+2
ISSP3=ISS+3
ISSP4=ISS+4
INEQ=0
DO 30 I=1,ISR
PI(I)=1.0
ZZZ=CAI(I)
CAI(I)=ALOG(ZZZ)-ETA(I)*ALOGT
CAIY(I)=CAI(I)
BETA(I)=0.0
XNU(I)=0.0
DO 30 J=1,ISS
BETA(I,J)=XNUJP(I,J)-XNUJ(I,J)
BET(I)=RF(I)+BETA(I,J)
XNU(I)=XNU(I)+XNUJ(I,J)
20 ZZZ=CMA/CRP
CSTA=5*ALOG(ZZZ)
SL64=SL/6468
C
COMPUTE AND TEST RANK OF BETA MATRIX
I=0
I=I+1
IF (I=ISS) 50,50,51
DO 70 J=1,ISR
IF (BETA(J,I)) 60,70,60
30 K=J
K=K
DO 90 M=1,ISS
BETA(K,M)=BETA(I,M)
BETA(K,M)=BETA(M,K)
90 BETA(I,J)=4B
I=I+1
DO 100 J=1,ISS
CA=BL(I,J)/BETA(I,J)
100 IF (CA) 110,110,110
DO 110 N=1,ISS
BETA(M,N)=BETA(M,N)-CA*BETA(N,M)
BETA(M,N)=BETA(M,N)-CA*BETA(N,M)
IF (ABS(BETA(M,N))<.01) 120,120,130
110 BETA(M,N)=0.0
120 CONTINUE
130 CONTINUE
GO TO 40
150 IF (IX=ISMC) 160,170,160
160 WRITE (6,1090) NNON 166
C ERROR EXIT NO. 1  ********************************************** NNON 167
C BETA MATRIX OF INCORRECT RANK NNON 168
IERR=1 NNON 169
GO TO 1000 NNON 170
C RESET BETA MATRIX (DESTROYED DURING RANK CALCULATION) NNON 171
170 DO 180 I=1,ISR NNON 172
DO 180 J=1,ISS NNON 173
180 BETA(I,J)=XNUIJP(I,J)-XNUIJ(I,J) NNON 174
AR=0.0 NNON 175
LG=0 NNON 176
DELT=DELT NNON 177
DO 200 I=1,ISC NNON 178
CI(I)=0 NNON 179
DO 190 J=1,ISS NNON 180
190 C(I)=CI(I)+GJA(J)*LPIJ(J,I) NNON 181
200 CONTINUE NNON 182
GO TO 220 NNON 183
C END OF INITIALIZATIONS NNON 184
210 IF (MODLPT) CALL NETMP (ITS,XMOD1,XMODEL) NNON 185
GO TO 230 NNON 186
220 CALL DENIS NNON 187
IF (ENR) RETURN NNON 188
IF (.NOT.FAILED) GO TO 230 NNON 189
C ERROR EXIT NO. 2  ********************************************** NNON 190
IENR=2 NNON 191
GO TO 1000 NNON 192
230 IF (XMSF+OR.AMACH.LT.1.5) GO TO 240 NNON 193
XMSF=.TRUE. NNON 194
XMOD1=XMP1 NNON 195
ITS=1 NNON 196
CALL NEXTMP (ITS,XMOD1,XMODEL) NNON 197
240 IF (IUDP) 250,350,250 NNON 198
C TESTS FOR SWITCH TO DOWNSTREAM REGION NNON 199
250 IF (OLLOGA) 450,260,260 NNON 200
260 IF (CX) 450,450,270 NNON 201
270 IF (AR) 300,280,300 NNON 202
280 AR=1+0 NNON 203
DO 290 I=1,NPRFLS NNON 204
290 OMBST(I)=I-DELBL(I) NNON 205
300 IF (APNX.LT.DATEST) GO TO 450 NNON 206
C SAVE DATA AT SWITCH POINT FOR POSSIBLE RESTART NNON 207
T8(1)=CX NNON 208
T8(2)=CT NNON 209
T8(3)=SC NNON 210
T8(4)=SCT NNON 211
T8(5)=DELTA NNON 212
T8(6)=HEDEL NNON 213
T8(7)=CET NNON 214
T8(8)=SCL NNON 215
T9(9)=CHA NNON 216
IT8(1)=INEQ NNON 217
IT8(2)=DNS NNON 218
IT8(3)=NNN NNON 219
IT8(4)=NROUT
DO 310 I=1,ISS
TB(I+10)=GJ(I)
BLKX(I)=RSA
IF (IS3B.EQ.0) GO TO 340
DO 320 I=1,10
320 BLKX(I+1)=BLNERR(I)
BLKX(12)=XBP
BLKX(13)=XPB
DO 330 I=1,NPRFLS
BLKX(I+3)=BLINT(I)
BLKX(I+5)=XSN(I)
BLKX(I+7)=DEBLB(I)
330 BLKX(I+19)=DDELB(I)
C SWITCH TO DOWNSTREAM REGION
340 CALL THROAT
IF (ERR) RETURN
GO TO 450
IF (DLGR) 450,450,360
360 WRITE (*,1100)
370 IF ((AFNX-DATEST)/DATEST.LT.0.05) GO TO 380
C ERROR EXIT NO. 3 ************************************************************************
C POSITIVE DLGR FAR BEYOND SWITCH POINT
ERR=3
GO TO 1000
C SWITCH TO DOWNSTREAM REGION WAS PREMATURE.
C INCREASE DATEST AND RESTART UPSTREAM SOLUTION AT SWITCH POINT.
380 DATEST=2.*((DATEST-1.)*1.)
IUPD=1
AR6=AR6+1.0
IF (AR6.LT.ARBA) GO TO 390
C ERROR EXIT NO. 4 ************************************************************************
ERR=4
GO TO 1000
390 TSZ=PRINT
TPRINT=0.C
CALL PRTA
IF (ERR) RETURN
ITAPE=NRCCONT-IWB(4)
IF (ITAPE.EQ.0) GO TO 410
JACKSPACE BINARY OUTPUT TAPE TO ELIMINATE RECORDS OF INVALID
C STEPS BEYOND SWITCH POINT
400 DO 400 I=1,ITAPE
C BACKSPACE ITPOUT
410 TPRINT=TSZ
C RESET DATA FOR RESTART AT SWITCH POINT
CX=TB(1)
CT=TB(2)
SC=TB(3)
CTT=TB(4)
DELTAX=TB(5)
HDELX=TB(6)
CTE=TB(7)
SCD=TB(8)
CHA=TB(9)
INEQ=ITB(1)
420    GJ(1)=TB(1+10)
430    BLNEX(I)=BLBK(I+1)
440    COPLU=FALSE.
450    CONTINUE
        IF (INCC) 610,460,610
460    PERTURBATION SOLUTION
        CALL PERT
        IF (ERR) RETURN
        IF (IUPD.EQ.0.AND.DELT.EQ.DELT1) DELT=DELT2
        C DETERMINE MAXIMUM AND MINIMUM PERTURBATIONS IN CHI(I)
        DCHMAX=CH
        DCHMIN=1.E30
        DO 480 I=1,15
        ADCH=ABS(SDCH(I))
        IF (ADCH>DCHMAX) 480,480,470
        DCHMAX=ADCH
        IMAX=I
480   WRITE(6,1110) DCHMIN,DCHMAX,IMAX
        IF (DCHMAX.EQ.0.) GO TO 580
        RATIO=DCHMIN/DCHMAX
        IF (RATIO<GE.DCHRAT) GO TO 510
        C EXCESSIVELY SMALL RATIO OF MINIMUM AND MAXIMUM PERTURBATIONS
        CH
        DO 500 J=1,15
        IF (BETA(IMAX,J)*PICHI(IMAX)*GE.0.) GO TO 500
        IF (G(J,J).GT.GAMIN) GO TO 500
        IF (IRA.LE.50) GO TO 490
        WRITE(6,1010)
500   GO TO 530
        C ARTIFICIAL INCREASE OF RATE CONSTANT TO PREVENT PREMATURE START
        OF INTEGRATION
        490   RATIO2=1.1*DCHRAT/RATIO
        CAI(IMAX)=CAI(IMAX)+ALOG(RATIO2)
        WRITE(6,1020) RATIO,HP(J),IMAX,G(J),IMAX,RATIO2
        IRA=IRA+1
        GO TO 220

500    CONTINUE
        C TESTS FOR SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION
510 IF (DCM.MAX.LT.CCHI) GO TO 580
   IF (DCM.MAX.LE.PCTEST.CCHI) GO TO 540
   MAXIMUM SDCHI(I) IS TOO LARGE -- BACKSTEP THE PERTURBATION
   SOLUTION TO FIND THE CORRECT SWITCHING POINT
   DELT=DELT/2.0
   ISUBD=ISUBD+1
   IF (ISURD.GT.23) GO TO 520
   CT=CT+DELT
   CTE=CT
   NAMELST /NEQDMP/ INEQ.IUPD.DLOGA.AR.CX.CT.DELT.AFNTS.
   IF (ISWB.NE.0) WRITE (6,NEQDMP)
   GO TO 600
520 WRITE (6,1120) ISUBD
530 WRITE (6,NEQDMP)
C ERROR EXIT NO. 5 ***************************************************
   IERR=5
   GO TO 1000
   C SWITCH FROM PERTURBATION SOLUTION TO INTEGRATION
   DXM=3.01*DCHMIN/DCHLL
   IF (DELTAX.LE.DXM) GO TO 550
   DELTAX=DXM
   HDEX=3.5*DELTAX
   T8(5)=DELTAX
   T8(6)=HDEX
   DXOLD=DELTAX
550 CT=CT+PCT
   CTE=CT
   RH0=RH0+PRHO
   GO TO 560 I=1,ISR
   IF (CAI(I).EQ.CAIV(I)) GO TO 560
   CAI(I)=CAIV(I)
   WRITE (6,103C) I
   WRITE (6,103C) I
   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 ***************************************************
   IERR=6
   GO TO 1000
   C CONTINUE PERTURBATION SOLUTION
   580 IF (CXL.T.XMODEL) GO TO 590
      MOLPT=TRUE
      CALL NEXTMP (ITS,XMOD1,XMODEL)
      C PRINT POINT IN PERTURBATION SOLUTION
   590 CALL PRTA
      IF (ERR) RETURN
      IF (CXL.EQ.XMAX) GO TO 660
      IF (CT.LE.TSTOP) GO TO 660
      ISUBD=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
510 IF (CT=1.0) 630, 630, 620
520 WRITE (6,1130)
C
ERROR EXIT NO. 7
ERR=7
IF (ERR) RETURN
GO TO 1000
630 IF (CX*GE.CXMAX) GO TO 660
IF (CT.LE.TSTOP) GO TO 660
NNN=NNN+1
PRINT POINT IN SOLUTION BY INTEGRATION
CALL PRTA
IF (ERR) RETURN
IF (NNN-QNS) 670, 640, 640
DELTA=XDELTAX
DELNL=DELTAX/2.0
NNN=NNN+1
NNN=0
GO TO 670
C
FINAL POINT
660 TPRINT=6.0
CALL PRTA
AXISYM=TRUE
CALL MODEL
WRITE (6,1160) ISTEPS
C
MAIN RETURN STATEMENT
RETURN
C
SAVE DATA AT BEGINNING OF INTEGRATION STEP FOR POSSIBLE RESTART
DO 660 I=1,155
GJB(I)=GJB(I)
680 CONTINUE
CXB=CX
CTJ=CT
CXL=CT
CAB=AFNX
RHO=RHO
CRS=CN
CH=CN
DELB1(1)=DELB1(1)
690 IF (NNR*GE.2) DELBLB(2)=DEBLB(2)
IF (CR=DELTAX*LTX+MODEL) GO TO 700
VOL=PLN=TPUP
VOL=DELTAX
HDXR=MODEL
DELTA X = XMODEL - CX
MDELX = 0.5 * DELTA X
700 IF (CX + DELTA X * LT + CXMAX) GO TO 710
DELTA X = CXMAX - CX
MDELX = 0.5 * DELTA X
C INTEGRATION STEP
710 CALL RKT
IF (ERR) RETURN
IF (FAILED) GO TO 740
IF (.NOT. MODLP) GO TO 720
DELTA X = DXSAVE
MDELX = DXSAVE
720 CONTINUE
C TESTS FOR VALID INTEGRATION STEP
IFAIL = 1
IF (ABS(SDT/CX) GT TTEST) GO TO 740
IF (.NE.EQ.1) GO TO 730
IFAIL = 2
IF (ABS(SDTE/CX) GT TTEST) GO TO 740
IFAIL = 3
IF (.NE.(SDCH/CH) GT TTEST) GO TO 740
IFAIL = 4
IF (.GT. LOGA(CXGE'] - 0.01)) GO TO 760
NIN = 463
C INTEGRATION STEP FAILED. REDUCE STEP SIZE AND RESTART AT
C BEGINNING OF STEP
740 DELTA X = DELTA X / SC
MDELX = DELTA X / 2 + 0
MODLP = .FALSE.
ICOUNT = ICOUNT + 1
IF (ICOUNT GT 30, OR DELTA X LE .1E-10) GO TO 990
IF (.GT. ICOUNT) = IFAIL
DO 750 I = 1, ISS
750 GJ(I) = GJR(I)
CT = CTN
CTE = CTEB
CX = CXB
AFNX = CARB
RHO = RHOB
SEN = CRS
CHA = CHD
DELRL(1) = DELRLB(1)
IF (NRFLS EQ 2) DELBL(2) = DELBLB(2)
NIN = 0
NNS = 0
SC = SC - 0.1
SC = AMAX1(SC, 1.1)
SC = SC - SC
NEG = 1
CALL DERIVS
IF (ERR) RETURN
IF (.NOT. FAILED) GO TO 690
C ERROR EXIT NO. 8
690 IF (X = X) GO TO 760
C FURTHER VALIDITY CHECKS ON STEP
760 DD 790 I = I, ISS
Caoa

FAIL=I+10
GJB(I) + GJB(I) + SDGJ(I)
IF (ABS(SDGJ(I))/GJB(I) + LE GEST) GO TO 770
IF (GJB(I) - GMIN) GO TO 740

770
C SUPPRESS REACTIONS INVOLVING STEP-SIZE CONTROLLING SPECIES
C WITH CONCENTRATION BELOW GMIN AND FALLING

NSUPR = 0
DO 780 K = 1, ISR
IF (NT*EDOT) GO TO 780
NOREAC(K) = 1, TRUE
NSUPR = NSUPR + 1
780
CONTINUE
WRITE (6, 1053) HP(I), GJB(I), (ISUPR(K), K = 1, NSUPR)
GO TO 740

790
C VALID STEP, APPLY INCREMENTS TO DEPENDENT VARIABLES.
C RESTORE CONSERVATION OF CHEMICAL ELEMENTS
DO 820 J = 1, ISC
CJ = 0
DO 810 I = 1, ISM
CJ + GJB(I)*LPIJ(I, J)
820
DO 830 I = 1, SS
GSO(I) = GSO(I) + 2
IF (GSO(I) > 900) OR (I GT ISC) GO TO 830
WRITE (6, 1060) I
C ERROR EXIT NO. 9 *********************************************
YEAR = 0
GO TO 1000
830
CONTINUE
ISMCP1 = ISMC + 1
DO 850 K = 1, ISC
DCA(K) = 0
DO 840 J = 1, ISC
DO 830
DCA(K) = DCA(K) + DCL(J)*AIC(J, K)
DCA(K) = DCA(K) + GSQ(K)
850
CONTINUE
DO 890 I = 1, ISC
I I = I + ISC
AA(I, ISMCP1) = 0
DO 860 K = 1, ISC
AA(I, ISMCP1) = AA(I, ISMCP1) + CDIJ(I, K) + DCA(K)
AA(I, ISMCP1) = AA(I, ISMCP1) + GSQ(I)
860
AA(I, ISMCP1) = AA(I, ISMCP1) + CDI J(I, K) + DCA(K)
DQ 880 C = 1, ISC
AA(I, ISMCP1) = AA(I, ISMCP1) + GSQ(I)
870
AA(I, L) = AA(I, L) + CDIJ(I, K) + CDIJ(L, K) / GSQ(K)
AA(I, L) = AA(I, L) + GSQ(I)
IF (I.EQ.L) AA(I,L)=AA(I,L)+ONE

CONTINUE

CALL D=SOL (AA, ISMC, 22)
IF (ERR) RETURN
DO 960 I=1, ISMC
I=I+1
DO 960 I=1, ISMC

GJ(I)=GJ(I)+AA(I, ISMC)
DO 930 K=1, ISMC
GJ(K)=GJ(K)+CI(J)*AIN(J,K)
DO 920 L=1, ISMC
IF (GJ(I) .LE. 740) GO TO 940
CONTINUE

IF (BAIL=1+30) RETURN

C COMPUTE SUPPLEMENTARY FLOW VARIABLES AND DERIVATIVES AT END OF
C STEP (1ND BEGINNING OF NEXT STEP)

CALL DERIVS
IF (ERR) RETURN
IF (I.EQ.1) GO TO 740
IF (I.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
I=5
DOEM=AMAX1(QTEST*ABS(OEOLDJ),DQMAX)
IF (ABS(QDPE=OEOLD))GT.DOEM) GO TO 740
DQMAX=AMAX1(DQMAX,DOEM)

QEOLD=QDPE

IF (IS5) GO TO 970
WRITE (6,1140) CX,DELTAX,TP,TEP,CHA,ODPE,ICOUNT
IF (ICOUNT.EQ.0) GO TO 960
WRITE (6,1040) (IF(K).K=1,ICOUNT)

ICOUNT=ICOUNT+1

ISTEPS=ISTEPS+1
DELTA=AMAX1(DELTA,0.7*DXOLD)
DXOLD=DELTA
GO TO 210

WRITE (6,1150)
C ERROR EXIT NO. 10
IERR=10
IF (IUPD.EQ.0) GO TO 370
WRITE (6,1070) IERR
CALL DUMP (RNAME)
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 (29H CHMIN/CHMAX=,1PE10.3,41H IS TOO SMALL CONCENTRATION NON 610
1ATION OF REACTANT,AA=,16H IN REACTION NO,,,13,3H IS,1E10.3,6HMOL/GMNON 611
2/36H =/= RATE CONSTANT FOR REACTION NO,,13,25H INCREASED BY A FACTOR 612
3CTOR OF,1E10.3,27H FOR PERTURBATION SOLUTION/) NON 614
1030 FORMAT (36H =/= RATE CONSTANT FOR REACTION NO,,13,27H RESTORED TO NON 615
1 CORRECT VALUE/) NON 616
1040 FORMAT (24H CAUSES OF STEP FAILURES,2014) NON 617
1050 FORMAT (28H CONCENTRATION OF,AA=,10H FROZEN AT,1PE10.3,9NON 618
1H MOLE/GM,11X,20H SUPPRESSED REACTIONS,3413/30X,3013) NON 619
1060 FORMAT (4H0GJ(,12,16H)**2 UNDERFLOWED/) NON 620
1070 FORMAT (19H ERROR EXIT NO,,13,11H FROM NONEQ/) NON 621
1080 FORMAT (3H1NONEQUILIBRIUM SOLUTION/) NON 622
1090 FORMAT (33H1DETA MATRIX OF INSUFFICIENT RANK/) NON 623
1100 FORMAT (18H DLOGR IS POSITIVE/) NON 624
1110 FORMAT (8H CHMIN=,1PE10.3,10X,7HCHMAX=,1E10.3,10X,5HIMAX=,12) NON 625
1120 FORMAT (55HBACKSTEPPING OF PERTURBATION SOLUTION TERMINATED AFTER NON 626
1,13,31H STEPS, DIAGNOSTIC DATA FOLLOW/) NON 627
1130 FORMAT (41H TEMPERATURE GREATER THAN RESERVOIR VALUE/) NON 628
1140 FORMAT (39H conv=,1PE12.5,5X,7HDELTA=,1E10.3,5X,2HT=,0PF8.2,5X,4HTEP=) NON 629
1150 FORMAT (60H =/= RATE CONSTANT INCREASED BY A CONVERGENT NON 630
1 STEP/) NON 631
1160 FORMAT (33H =/= RATE CONSTANT INCREASED BY A CONVERGENT NON 632
1,15,18H INTEGRATION ST/) NON 633
1EPS/) NON 634
END NON 635-
SUBROUTINE NRMAX  NRM 1
LOGICAL ERR  NRM 1
REAL ACOM(30), ELEMENT(10), HPI(20), NRM 1
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ  NRM 1
COMMON AA(22, 24), CDIJ(20, 10), CAPX(20)  NRM 1
COMMON 6, AFNTS, AFN, AMACH, AR, ARBA, ARBB, B7F, NRM 1
C, CAR8, CH, CHA, CHB, CM, CMA, CPA, CRP, NRM 1
CRR8, RS, CSTA, CT, CTAP, CTB, CTT, CTMAX, CTMXX, NRM 1
CTP, CL, CTT, CX, CXS, CXA, DATES, DBTEST, DELT, NRM 1
DELTA, DELTA, DLGGA, DLOGR, DT, ENT, FLUX, HDELX, NRM 1
PREST, PRMC, RHAP, RHQ, RHOB, NRM 1
RMFH, RHDC, RHOP, RMPL, RHTH, ROBARA, ROBAP, NRM 1
SCPG, SDL, NRM 1
SEN, SHPG, SC, SL, SL64, SM, SU, SU2, SUM, NRM 1
TEST, TPRINT, TSSTOP, U, ZP, ZPA, NRM 1
COMMON BE(1), BET(20), BLBK(31), CAI(64), CAPXTH(20), NRM 1
CEACT(64), CGI(20), CGMU(20), CHI(64), CHII(20), NRM 1
CLNMC(64), CLNP1(64), CMW(20), ETAI(64), ETAJ(20), GJA(20), NRM 1
GJII(20), PEHTGJ(20), PGJ(20), PII(64), PICH(64), GM(20), NRM 1
GO(64), SAJ(20), DOCHI(64), SENT(20), SHJ(20), SHJAI(20), NRM 1
SKILL(20), SS(20), TB(30), TFA(20), TFB(20), TFC(20), NRM 1
TFD(20), TFE(20), TK(20), THEV(20), XN settings(20), XNUI(64), NRM 1
COMMON BETA(64, 20), ELJ(10, 20), GELJ(10, 20), NRM 1
XNU11(64, 20), XNUI1P(64, 20), NRM 1
COMMON IC, IM, INEQ, INEV, IP, IRUN, ISC, ISCP1, NRM 1
ICMC, ISMCNR, ISI, ISS, ISSNR, ISSP1, ISSP2, ISSP3, ISSP4, NRM 1
ISW1A, ISW1B, ISW2A, ISW2B, ISW3, ISW4, ISW4B, ISW5A, NRM 1
ISW5B, ISW6A, ISW6B, IUPO, IZERO, JK, LC, M1, NFIT, NRM 1
NIT, NNN, NNS, NOS, NOT, NTTEST  NRM 1
COMMON IGJ(20), IGMO, ITB(5), KUR(64, 20), LPI(20, 10), NRM 1
COMMON ACOM, ELEMENT, HP, NRM 1
DIvision AA(22, 24), BTA(64, 20), CAPQ(31), CCI(20), DJG(20), GJ(20), SBJ(20), NRM 1
JL(20), SDGJ(20), SHJAP(20), THEV(20), NRM 1
EQUIVALENCE (AA(1, 1), AA(1, 1)), (BTA(1, 1), BTA(1, 1)), NRM 1
(CBLK1, CAPHQ(1), (CAPXTH(1)), CCI(1)), (GJA(1), DJG(1)), NRM 1
(CAPX1, GJ(1)), (SAJ(1), SBJ(1)), (SJ(1), SDGJ(1)), NRM 1
(SHJAI(1), SHJAP(1)), (THEV(1), THEV(1)), NRM 1
COMMON /ERR/ ERR  NRM 1
M1=ISC+1  NRM 1
DELT=DELTI  NRM 1
PRE=1+0  NRM 1
TI=1+0  NRM 1
CT=TI  NRM 1
00 10 J=1, ISS  NRM 1
C(J)=GJA(1)  NRM 1
FI=0  NRM 1
CT=CT-DELT  NRM 1
T2=CT  NRM 1
CALL NEWRAP  NRM 1
IF (ERR) RETURN  NRM 1
F3=FLUX  NRM 1
30 CT=CT-DELT  NRM 1
T3=CT  NRM 1
CALL NEWRAP  NRM 1
IF (ERR) RETURN  NRM 1
F3=FLUX
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, 100)
RETURN

1H A P T 31 (I) 6 7 E 8 3 7 1 9 2 4 5 6 9 3 1

100 FORMAT (22H DEFINITIONS OF SYMBOLS)"
FROZEN SHOCK (ATM)/50H PWW = SURFACE PRESSURE ON A WEDGE MODEOUT 56
SL (ATM) OUT
FROZEN SHOCK = STAGNATION-POINT HEAT FLUX TIMES SORT(R) FOOUT 58
1H EQUILIBRIUM SHOCK ON A FLAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDARY LAYER/20X.40H(BTU/ST-SEC)FT\*0.5 59
2DARY LAYER/20X.40H(BTU/SL) ST-SEC)FT\*0.5 60
3F = STAGNATION-POINT HEAT FLUX TIMES SORT(R) FOR EQUILIBRIUM SOU 61
4SHOCK ON A FLAT-FACED 4"" WITH A FROZEN BOUNDARY LAYER/20X.40H(BTU/ST-SEC)FT\*0.5 62
5U/SQ FT-SEC)FT\*0.5 FAY-RI DELL/127H QEHE = STAGNATION-PO 63
6INT HEAT FLUX TIMES SORT(R) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICOU 64
7AL MODEL WITH AN EQUILIBRIUM BOUNDARY LAYER/20X.40H(BTU/ST-SEC)FT\*0.5 65
8FT\*0.5 FAY-RI DELL/127H QEHE = STAGNATION-POINT HEAT FLUX FOOUT 66
95 SORT(R) FOR EQUILIBRIUM SHOCK ON A HEMISPHERICAL MODEL WITH A FO 67
1OZEN BOUNDARY LAYER/20X.41H(BTU/SQ FT-SEC)FT\*0.5 68
SHELL) OUT 69
EQUILIBRIUM SHOCK ON A MODEL (BTU/SQ FT-SEC)FT\*0.5 SRI FORMULAOUT 70
3/125H QFEHE = STAGNATION-POINT HEAT FLUX TIMES SORT(R) FOR FRO 71
4H SHOCK ON A FLAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDARY LAYER/20O 72
50X.41H(BTU/ST-SEC)FT\*0.5 FAY-RI DELL/119H QFEHE = STAG 73
6GATION-POINT HEAT FLUX TIMES SORT(R) FOR FROZEN SHOCK ON A FLAT-FO 74
7ACED MODEL WITH A FROZEN BOUNDARY LAYER/20X.41H(BTU/SQ FT-SEC)FT\*OUT 75
955 FAY-RI DELL) OUT 76
FROZEN SHOCK ON A HEMISPHERICAL MODEL WITH AN EQUILIBRIUM BOUNDAR 77
2R LAYER/20X.40H(BTU/SQ FT-SEC)FT\*0.5 FAY-RI DELL/122H QFHFOUT 78
3 = STAGNATION-POINT HEAT FLUX TIMES SORT(R) FOR FROZEN SHOCK ONOUT 79
4H SHOCK ON A FLAT-FACED MODEL WITH AN EQUILIBRIUM BOUNDARY LAYER/20OUT 80
5X.41H(BTU/ST-SEC)FT\*0.5 FAY-RI DELL/79H QFHFOUT 81
6T PLATE 1 FT FROM THE LEADING EDGE (BTU/SQ FT-SEC) OUT 82
FROZEN SHOCK ON A MODEL (BTU/SQ FT-SEC)FT\*0.5 SRI FORMUL 83
4H QAD = RADIATED POWER (CAL/CM FT-SEC)/51H QW = HEAT FLUX OUT 84
3 TO NOZZLE WALL (BTU/SQ FT-SEC)/64H QWW = HEAT FLUX TO SURFACEOUT 85
4 OF A WEDGE MODEL (BTU/SQ FT-SEC)/44H R = DENSITY IN FREE STOUT 86
STREAM (LB/CU FT)/50H REP = REYNOLDS NUMBER FOR FOOT IN FREE STOUT 87
6HEAM/54H RETH = REYNOLDS NUMBER BASED ON MOMENTUM THICKNESS/95H OUT 88
7EHTMTR = CRITICAL REYNOLDS NUMBER (BASED ON MOMENTUM THICKNESS) FOOUT 89
8 FOR BOUNDARY LAYER TRANSITION/75H RTE = DENSITY AT STAGNATION COUT 90
9NFORMATION FOR EQUILIBRIUM SHOCK (LB/CU FT)/48H S = ENTROPY INOUT 91
% FREE STREAM (BTU/LB-DEG R)/56H SIGMA = ELECTRICAL CONDUCTIVITY OUT 92
$IN FREE STREAM (MHO/CM) OUT 93
FOZEN SHOCK (MHO/CM)/54H SIGT2F = ELECTRICAL CONDUCTIVITY AT STAGNATION CONDOUT 94
1ON FOR EQUILIBRIUM SHOCK (MHO/CM)/54H SIGT2F = ELECTRICAL CONDUCTOUT 95
2TIVITY AT STAGNATION CONDITION FOR FROZEN SHOCK (MHO/CM)1/49H STANDOUT 96
3N = STANTON NUMBER BASED ON TOTAL ENTHALPY/79H ST2F = ENTROPY OUT 97
4AT STAGNATION CONDITION FOR EQUILIBRIUM SHOCK (BTU/LB-DEG R)/74H SOUT 98
5T2F = ENTROPY AT STAGNATION CONDITION FOR FROZEN SHOCK (BTU/LB-OUT 99
6DEG R)/45H T = TEMPERATURE IN FREE STREAM (DEG K)) OUT 100
1TERM FOR EQUILIBRIUM SHOCK (DEG K)/71H T2F = TEMPERATURE AT STAGNATION CONDOUT 101
LATION FOR FROZEN SHOCK (DEG K)/55H T = TEMPERATURE AT STAGNATION CONDOUT 102
3E WALL (L3F/ST FT)/39H TELEC = ELECTRON TEMPERATURE (DEG K) OUT 103
4THETA = MOMENTUM THICKNESS OF BOUNDARY LAYER ON NOZZLE WALL (INOUT 104
5H)/43H V = VELOCITY OF FREE STREAM (FT/SEC)/57H #IDTH = FOOUT 105
FIRST TRANSVERSE DIMENSION OF CHANNEL (INCHES)/93H X = AXIAL DOUT 111
71STANCE ALONG NOZZLE, MEASURED FROM THROAT AND POSITIVE DOWNSTREAM 112
8 (INCHES)/67H XSN = N AVERAGED OVER SEVERAL PRECEDING POINTS DOUT 113
OF THE SOLUTION/66H XW = DISTANCE FROM LEADING EDGE OF WEDGE, DOUT 114
$MEASURED ALONG WEDGE SURFACE (INCHES)/11SH YS = ORDI NATE OF SOUT 115
SHOCK FRONT LINE PARALLEL TO FREE STREAM FLOW, PASSING THROUGH LEAD DOUT 116
SHING EDGE OF WEDGE (INCHES)/71H ZETA = NONDIMENSIONAL STREAMWISE DOUT 117
COORDINATE IN BLUNT WEDGE ANALYSIS) OUT 118
END OUT 119-
SUBROUTINE OUIT
OUTPUT ROUTINE

LOGICAL, ERP, DATAPE, SUPGO, NOTRAN
REAL ACON(30), ELMNIT(10), HPA(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22, 24), CDIJ(20, 10), CAPX(20)

COMMON C, AFNIS, AFNX, AMACH, AR, ARBA, ARBB, BZERO, CBD

COMMON C
DATA COMMON, VARNAM, WNAME, WIDTH, ANAM, ARAT

END
FAREA=2.*RTHCM(1)
GO TO 100
IF (NPRFLS.EQ.1) GO TO 80
DO 70 I=1,2
DELSTP(I) = (1.-OMDST(I))*RO
FAREA=4.*[RTHCM(1)-DELSTP(I)]*([RTHCM(2)-DELSTP(2)]
GO TO 100
DO 10 I=1,2
FAREA=PPZ*(OMDST(I)*RTHCM(1)***2
FAREA=FAREA/929.03
WRITE (6,570)
TFLOW=FAREA*CF(1)*CF(2)*SM
IF (ISM2B.GT.0) GO TO 110
WRITE (6,580) FLOW
IF (ISM2B.EQ.0) GO TO 120
DEL=SAVEC(I)*RHO*TB7/CW
GO TO 130
DEL=0.
IF (NPRFLS.EQ.2) GO TO 140
WRITE (6,590) FACNAM,DIAM(1)
GO TO 150
WRITE (6,600) CHANAM,DIAM(1),DIAM(2),FACNAM
WRITE (6,610) POUT,TOUT,HOUT,SOUT,ROUT,OUT,FLX,TFLOW,GAMMA,CW,DEL
WRITE (6,620)
WRITE (6,630) (HP(I),SAVEC(I),I=1,ISS)
RETURN

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*3B.EQ.0) GO TO 180
DO 170 I=1,NPRFLS
II=7*10*I
FVOUT(II)=CF(6)*DEBL(I)
FVOUT(II+1)=CF(9)*THETA(I)
FVOUT(II+2)=QWODT(I)
FVOUT(II+3)=TAUW(I)
FVOUT(II+4)=1.*BHR
170 FVOUT(II+5)=PRREF
180 CONTINUE
VARNAM(6)=DNAM
VARNAM(6)=ANAM
IF (NPRFLS.EQ.1) GO TO 200
CALL ГVAR2 (CX,Y,Z)
IF (ERR) RETURN
IF (ISM2B.NE.0) GO TO 190
CHDIMS(2)=2.*Y*CF(4)
CALL GARM(X*"")
IF (E2H) RETURN
FVOUT<6)=2.*Y*CF<4)
IF (N7TPAN) GO TO 220
CALL TYANSP(FVOUT(~)*FVOUT(~))
IF (ERR) RETURN
FVOUTC14)=VISC*CF(3)
IF (VISC*EQ*0.) GO TO 230
FVOUT(12)=FVOUT(5)*FVOUT(7)/FVOUT(14)
GO TO 240
230 FVOUT(12)=0.
240 FVOUT(15)=SIGMA
GO TO 250
250 FVOUT(12)=SIGMA
FVOUT(14)=CF(3)*VISC.
FVOUT(15)=SIGMA
260 FVOUT(13)=CM
H3)=1.8*CHA*CRP/CMA
IF (ISw3B*EQ*0.) GO TO 340
IF (VOUT<7)+EQ*0.) GO TO 270
STFAC=1./(FVOUT(5)*FVOUT(7)+(H0-HW))
270 FVOUT(23)=STFAC*FVOUT(19)
FVOUT(24)=FVOUT(12)*FVOUT(18)/12.
FVOUT(25)=20.*EXP(-0.224*FVOUT(8))
IF (NPRFLS*EQ*1.) GO TO 280
FVOUT(33)=STFAC*FVOUT(29)
FVOUT(34)=FVOUT(12)*FVOUT(28)/12.
FVOUT(35)=FVOUT(25)
280 IF (LIMOUT*EQ*0.) GO TO 300
290 FV02(1,1)=SN(I)
FV02(2,1)=SN(1)
CALL GEMAR(CX*ARAT*ARAT)
IF (ERR) RETURN
IF (NPRFLS*EQ*2) GO TO 310
FVOUT(16)=ARAT
GO TO 320
310 FVOUT(6)=ARAT
VARN4(6)=ANAM
GO TO 340
320 IF (JDM*EQ*0.) GO TO 330
FVOUT(6)=RTHCM(1)*SORT(ARAT)*CF(4)*2
GO TO 340
330 FVOUT(6)=RTHCM(1)*ARAT*CF(4)*2.
340 INEQP1=INEQ+1
WRITE (6,640) (ASTRSK,I=1,8), (ASOLN(J=1,CLN),J=1,3), (ASTRSK,K=1,3) OUT 221
1.*NSTEPS,STEP(ISCLN), (ASTRSK,L=1,4), TYPSSLN(INEDP1.ISDNN) OUT 222
WRITE (6,650) (VARNAM(I),FVOUT(I),I=1,15) OUT 223
IF (ISW3B.EQ.0) GO TO 360
DO 350 L=1,NPRFLS
11=6+10*L
12=11+9
WRITE (6,660) (VARNAM(I),FVOUT(I),I=11,12) OUT 226
CONTINUE OUT 227
360 IF (NPRFLS.EQ.0 OR ISW3B.NE.0) GO TO 370
WRITE (6,670) CHDIMS(1), CHDIMS(2) OUT 228
WRITE (6,680) (VARNAM(I),TNOUT(I),I=1,4) OUT 229
IF (ISW6B.EQ.0 OR ISOLN.NE.1) GO TO 440
IF (16*LT.ISOL) GO TO 440
IS=0
WRITE (6,690) IF (IC*EQ.0) GO TO 400
EMF=SAVEC(I)
SAVEC(I)=EMF*RHO*TB7/CM
ISSSP1=(ISS=11/541
DO 410 K=1,ISS
LIM1=K*(K-1)+1
LIM2=4*IN(5K,ISS)
WRITE (6,700) (HP(I),SAVEC(I),I=LIM1,LIM2)
IF (IC*EQ.0) GO TO 420
SAVEC(I)=EMF
WRITE (6,710)
WRITE (6,510) [1,PI(I),I=1,ISR]
WRITE (6,520) [1,CHI(I),I=1,ISR]
WRITE (6,530) [1,CH(I),I=1,ISR]
DO 430 I=1,ISS
DLCDX(I)=GJ(I)
WRITE (6,550) (HP(I),DLCDX(I),I=1,ISS)
IF (*NOT*DATAPE) GO TO 500
IF (ITYPER.NE.1) GO TO 450
IMP=0
IX=0
IX=IX+1
NRCOUT=NRCOUT+1
ITYPER=2
XXX(1)=FVOUT(1)
XXX(2)=FVOUT(2)
XXX(3)=FVOUT(4)
XXX(4)=FVOUT(5)
XXX(5)=FVOUT(7)
XXX(6)=FVOUT(3)
XXX(7)=FVOUT(8)
IF (ISW3N) 460,480,460
460 IBL=17+10*(NBL-1)
XXX(9)=FVOUT(1BL)
IF (NPRLS.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 (1TPOUT) ITPY,IX,XXX(I),I=1,10
RETURN
C
C
510 FORMAT (57X,18HREACTION RATE DATA)
520 FORMAT (3H P-.11X,6(I4,1PE10.1),6X)
530 FORMAT (4H CH1,10X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)
540 FORMAT (6H PICH1,10X,8(I4,1PE10.1)/9(I4,1PE10.1),6X)
550 FORMAT (4H DLG,12X,7(2X,A4,1PE10.1),8(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 (1H GAS FLOW RATE =.F10.3,2X,7H LB/SEC)
590 FORMAT (1H NOZZLE =.A4,5X,F10.3,2X,21H INCH THROAT DIAMETER)
600 FORMAT (9H CHANNEL ,A4,5X,F10.3,0.3H BYF13,17H INCH THROAT FOR .A4)
610 FORMAT (1H9)
610 FORMAT (1H PRESSURE =.F1.0,3,2X,4H ATM/19H TEMPERATURE
1 =.F7.9,5X,6H CEG K/19H ENTHALPY =.F7.0,5X,7H BTU/LB/T9H HEAT)
2=IF6.2,3X,13H BTU/LB/SEC/19H DENSITY
3=IF6.2,5X,9H LB/CU FT/19H VEL OCITY
4=IF6.2,5X,7H FT/SEC/19H DENSITY
5=IF6.2,1X,26H LB/SEC/19H COMPUTED FLOW
6=IF6.2,3/1X,26H LB/SEC/19H MAGMA
7=IF6.2,3/1X,26H LB/SEC/19H ELECTRON DENSTY
8=IF6.2,3/1X,26H LB/SEC/19H ELECTRON DENSTY
9=IF6.2,3/1X,26H LB/SEC/19H ELECTRON DENSTY
10=IF6.2,3/1X,26H LB/SEC/19H ELECTRON DENSTY
2=IF6.2,3/1X,26H LB/SEC/19H ELECTRON DENSTY
70HS/CC1
620 FORMAT (10X,22HSETIES MOLE FRACIONS/)
630 FORMAT (13X,4X,1PE10.3)
640 FORMAT (10X,22HSPECIES MOLE FRACIONS/)
650 FORMAT (10X,22HSEPIES MOLE FRACIONS/)
660 FORMAT (10X,22HSEPIES MOLE FRACIONS/)
670 FORMAT (9H WIDTH =.F11.3,6X,9H HEAT =.F11.3)
690 FORMAT (10X,22HSETIES MOLE FRACIONS IN THE FREE STREAM)
700 FORMAT (5(1X,.*4,3X,1H=,4X,1PE10.3,3X))
FUNCTION PIOMEG(TE)  
COMMON /TNEQ/ TLIST(30), POM(30)  
DO 10 I=2,30  
IF (TE LT TLIST(I)) GO TO 20  
10 CONTINUE  
PIMEG=POM(30)  
GO TO 30  
20 PIMEG=POM(I-1)+(POM(I)-POM(I-1))*(TE-TLIST(I-1))/(TLIST(I)-TLIST(I-1))  
30 RETURN  
END
SUBROUTINE PROP
REAL ACOM(30),ELMEN(10),HP(20)
DOUBLE PRECISION AA,AAP,CAPX,GJ,CDIJ
COMMON AA(22,24),CDIJ(20,10),CAPX(20)
COMMON A*,AFN,N,AFNS,AMACH,AR,ARB,ARBA,ARB,BZERO
1 COMMON.0,j CARB,CH,CHA,CLNT,CM,CMA,CRB,CRP,
2 COMMON, C,CF,CF*,CFS,CSA,CT,CPT,CSC,CTB,CTE,CTMAX,CTMXK
3 C,CP,CTP,CTT,CX,CX**,CXB,CMAX,DATEST,DTTEST,DELT1
4 COMMON, CL,CR,DELT2,DELTAX,DLDTA,DLROA,DT,ENT,FLUX,HDELX,PCT
5 COMMON, FLUX*,HP*,HPBAR*,HPBAR,HPBAR,SUPR*,HBAR*,TEST*,TPRINT
6 COMMON, HBAR*,HP*,HPBAR*,HPBAR,HPBAR,HPBAR*,HPBAR,HPBAR*
7 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
8 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
9 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
10 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
11 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
12 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
13 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
14 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
15 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
16 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
17 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
18 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
19 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
20 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
21 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
22 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
23 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
24 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
25 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
26 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
27 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
28 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
29 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
30 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
31 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
32 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
33 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
34 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
35 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
36 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
37 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
38 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
39 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
40 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
41 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
42 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
43 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
44 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
45 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
46 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
47 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
48 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
49 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
50 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
51 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
52 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
53 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
54 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
55 COMMON, HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*,HPBAR*
IF (ERR) RETURN
NSTEPS=0
IF (.NOT.MODLPT) GO TO 30
CALL MODEL
MODLPT=FALSE
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 WRITE
DIMENSION OM(3)
COMMON /TRANS/ T(3,20,20),ZM2(20)
COMMON /TRANS7/ V(400)*KO(100),NQ(400),IQ(400),NKO
COMMON /TRANS8/ NV(15),N
COMMON /RDR/ ISW8B
DATA IDBG=-1/
IDBG=IDBG+1
WRITE(*,90) WRITE=FALSE.

C 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)
      CONTINUE
   C COMPUTE CROSS SECTION VALUES AND ACCUMULATE IN Q ARRAY.
      M=2
      LQ1=1
      DO 20 J=1,N
         K=Q(L)(L)
         LQ2=NQ(L)
         GO TO (20,30,40,50,60,70,80,90,100,110,120,130,140,150,160), K
   20 CONTINUE
      GO TO 190
      OM(1)=V(M)*Q(1,2,1)
      OM(3)=OM(1)*2.5625
      OM(2)=V(M+1)*OM(1)
      CALL QEXP (OM,V(M))
      CALL QTAB (OM,V(M))
      CALL QGIP (OM,V(M))
      CALL QGIO (OM,V(M))
      CALL QATTP (OM,V(M))
      CALL Q13 (OM,V(M))
      CONTINUE
      CALL QATTP (OM,V(M))
      CALL Q13 (OM,V(M))
      CONTINUE
      CALL QATTP (OM,V(M))
      CALL Q13 (OM,V(M))
      GO TO 190
      CALL QSAME (OM,V(M))
      GO TO 190
      CALL QMX (LQ1,LQ2)
      GO TO 210
      CALL Q12 (OM,V(M))
      GO TO 210
      CALL Q13 (OM,V(M))
      GO TO 210
      CALL Q13 (OM,V(M))
      GO TO 210
      CALL Q13 (OM,V(M))
      GO TO 210
      CALL Q13 (OM,V(M))
      GO TO 210
CALL Q14 (LQ1,LQ2,V(M))
GO TO 210
160 CONTINUE
GO TO 210
170 DO 180 LQ=LQ1,LQ2
I=IQ(LQ)
J=JQ(LQ)
Q(I+1,J)=OM(1)
180 Q(3,I,J)=OM(3)
GO TO 210
190 DO 200 LQ=LQ1,LQ2
I=IQ(LQ)
J=JQ(LQ)
DO 200 KK=1,3
200 Q(KK+1,J)=Q(KK+1,J)+OM(KK)
210 M=M+NV(K)
C DEBUG OUTPUT
IF (ISWB.EQ.0) GO TO 240
IF (IDBG.EQ.0) GO TO 220
IF (IDBG.NE.ISWB) GO TO 240
220 WRITE(6,250) L*K,NO(L),M,LQ1
230 DO 230 I=1,N
240 LQ1=LQ2+1
IF (WRITE) IDBG=0
RETURN
C
250 FORMAT (1HC/10I5/)
260 FORMAT (1P10E12.4)
END
SUBROUTINE OCOUL (OM1, X1, C1)
SUBROUTINE QCOUL COMPUTES THE FACTOR O8*QC IN THE EFFECTIVE
Coulomb cross section.

COMMON /TRANSL/ TeQ(3*20-20), ZM2(20)

RETURN
END

OM1 = 5*OM1 + ALCH(OM1)
RETURN
END
SUBROUTINE QEX (OM, V)

SUBROUTINE QEX COMPUTES CROSS SECTIONS FOR A RESONANT EXCHANGE

PROCESS

DIMENSION OM(3), V(3)

COMMON /TRANSL/ T, O(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 + .03724 * B ** 2

OM(3) = 2 * (Y - 4 * 317 * B) ** 2 - .06754 * B ** 2

OM(2) = 0

RETURN

END
SUBROUTINE QINTRP (A,B,N1,N)

SUBROUTINE QINTRP INTERPOLATES TABULAR DATA FOR THE CROSS SECTION COMPUTATIONS
COMMON /ROTR/ ISW8B
DIMENSION B(N)
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
30 IF (ISW8B.EQ.0) GO TO 40
WRITE (6,60) I,A,TL(I),OM(I+1)
40 DO 50 J=1:3
50 B(J)=OM(I+1,J)+(A-TL(I+1))*(OM(I+1,J)-OM(I+1,J))/((TL(I)-TL(I+1))
RETURN
C
60 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 QMIX (LQ1,LQ2)  
SUBROUTINE QMIX COMPUTES CROSS SECTIONS FROM THE EMPIRICAL MIXING RULE

DIMENSION SQT(3,20)
COMMON /TRANS1/ T,O(3,2G,20),ZM2(20)
COMMON /TRANS7/ V(4G0),KQ(100),NO(100),IQ(400),JQ(400),NKG
IF (LQ2.LT.LQ1) GO TO 70
DO 10 I=1,20
  SQT(I,1)=0.
10 DO 60 LQ=LQ1,LQ2
  I=IQ(LQ)
  J=JQ(LQ)
  IF (SQT(I,J).GT.C)* GO TO 30
  DO 20 K=1,3
    SQT(K,I)=S*SQRT(Q(K,I,J))
20  DO 30 K=1,3
    SQT(K,J)=S*SQRT(Q(K,J,J))
30  DO 40 K=1,3
    Q(K,I,J)=(SQT(K,I)+SQT(K,J))*2+Q(K,I,J)
40  RETURN
70  END
SUBROUTINE QREPP (QM, VV)

SUBROUTINE QREPP COMPUTES CROSS SECTIONS FOR AN INVERSE POWER LAW INTERACTION POTENTIAL.

DIMENSION QM(3), VV(3)

COMMON /TRANS1/ T, Q(3*20, 20), ZM2(20)

COMMON /TRANS4/ TL(1000), OMEGA1(1000), ASTAR(1000), BSTAR(1000)

ITL=VV(1)

OM(1)=OMEGA1(ITL)*T**(-2.0/VV(2))

OM(2)=STAR(ITL)*OM(1)

OM(3)=BSTAR(ITL)*OM(1)

RETURN

END
SUBROUTINE QSAME (CM, VV)
SUBROUTINE QSAME SETS THE CROSS SECTION EQUAL TO THOSE FOR ANOTHER PAIR.
DIMENSION CM(3), VV(5)
COMMON /TRANSI/ T,O(3,20,20), ZM2(20)
I=VV(1)
J=VV(2)
CM(1)=C*Q(1,I,J)
CM(2)=C*Q(2,I,J)*VV(4)
CM(3)=C*Q(3,I,J)*VV(5)
RETURN
END
SUBROUTINE OTAB (OM, V)

C USE TABULAR DATA FOR CROSS SECTIONS.
DIMENSION CM(3), V(3)
COMMON /TRANS1/ T, G(3, 20, 20), ZM2(20)
A = V(1)
N1 = V(2)
NL = V(3)
CALL QINTERP (T, CM, N1, NL)
OM(1) = A * CM(1)
OM(2) = A * CM(2)
OM(3) = OM(3) * OM(1)
RETURN
END
SUBROUTINE Q11 (LO1, LO2, VI)
SUBROUTINE Q11 MULTIPLIES CROSS SECTIONS BY A RAMP FUNCTION OF
TEMPERATURE.
DIMENSION VV(3), VI(2)
COMMON /TRANS1/ T0(3,20,20), ZM2(20)
DATA VV/3*1./
T0=VI(1)
T1=VI(2)
VV(1)=AMAX1(0.,(T-T0)/(T1-T0))
IF (VV(1).LT.1.) CALL Q14 (LO1, LO2, VV)
END
SUBROUTINE Q12 (OM,VV)
SUBROUTINE Q12 COMPUTES CROSS SECTIONS FROM THE GENERALIZED MIXING
C RULE (E0, 27)
C
DIMENSION OM(3),VV(4)
COMMON /TRANS/ Q(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),NQ(100),IQ(400),JO(400),NKQ

K1=VV(1)
K2=VV(2)
C=VV(3)

DO 10 LO=L01,L02
  I=IQ(LO)
  J=JO(LO)
  O(K1,I,J)=C*O(K2,I,J)
  RETURN
END
SUBROUTINE Q14 (L01, L02, 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 = l01, l02
i = iq(l0)

j = jn(l0)
Q(1, i, j) = vv(1) * Q(1, i, j)
Q(2, i, j) = C2 * Q(2, i, j)
Q(3, i, j) = C3 * Q(3, i, j)

10 continue

return

end
SUBROUTINE RADIUS (ITYPE, XR, AG, AGJ, L)
DIMENSION YZ(2), NPR(2)
COMMON /BL/ DELBL(2), BLINT(2), XZERO, TWALL, CPWALL, VISROT, DIAM(2),
1 SW, RO, JOIM, IPOINT
COMMON /BLRAD/ YOZO
COMMON /AREA/ ATPI(12), PARAM(3, 12, 2), RTHCM(2), NSECT(2),
1 NSECTU(2), ISHAPE(12, 2), NPROFL(2), NPRFLS, NBL
DATA NPR /2, 1/
GO TO (10, 10, 4C), ITYPE
10 CALL GMAR (XR)
RATIO=R/R0
GO TO (20, 30, 30), ITYPE
20 AG=RATIO
AGJ=1
GO TO 50
30 AG=RATIO**2
AGJ=AG
GO TO 50
40 CALL GMAR2 (X, YZ(1), YZ(2))
AG=YZ(1)*YZ(2)/YOZO
R=YZ(L)
M=NPR(L)
AGJ=YZ(M)**2
50 RETURN
END
COMMON /POLYAT/ THEVE(4,20)
COMMON /RDLIST/ IGAS, IGASO, NOZZO, ICHANO
COMMON /ROMAIN/ HS, SUPGO, MPITER, NOTRAN
COMMON /ROMOD/ LEWIS, IAMBIP
COMMON /RROUTE/ FLOW, PCNAM, CHANAM, LIMOUT
COMMON /ROTVF/ ISW9B
COMMON /ROWEDG/ ANGLE(10), RADL(5), WX1, WX2, WX(20), TWEDGE, W X

COMMON /PREDAT/ PFSAI, DELTHI, TPRNTI, DELTUXI, CMAXI

1 CTAPI, XZERCI, PARAM(3,12,2), HZERD1
2 HSTAG, READKS, READAG, ANNS, AXMOD
3 ICASE, NEELS, ISW7D, INT, ICHAN, NOSI, INEQV1, NREC0, JCF(10)
4 ISCI, ISSI, ISRI, ICII, IE(10), IR(64), NSECT(2,2), IEEP(10)

COMMON /RNEVE/ GAMIN, HTEST, TEHTST, QTEST, DCMH, OCHAY, CCHY, TTEST

1 GTEST
2 COMMON /SPEC/ SP1(43), SP2(43), SP3(43), SP4(43), SP5(43), SP6(43)
3 SP7(43), SP8(43), SP9(43), SP10(43), SP11(43), SP12(43), SP13(43)
4 SP14(43), SP15(43), SP16(43), SP17(43), SP18(43), SP19(43), SP20(43)
5 SP21(43), SP22(43), SP23(43), SP24(43), SP25(43), SP26(43), SP27(43)
6 SP28(43), SP29(43), SP30(43)

COMMON /TAPDOT/ XXX(10), ITPDUT, NRCOUT, IFLOW, MYPER, IMP, DATAPE

COMMON /THRT/ RSA

1 COMMON /TNE/ SUMGH, SCPGM, QDRP, QDPE

COMMON /TNE/ TN1(186), TN2(186)

COMMON /TNEQ/ TLIST(30), PCM(30)

1 COMMON /TNMOD/ CTEQ, DETP, EPAR(2,25), NTR, ITR(25), KTF(25)

1 KTR(25), ICM(25), IPA(25)

1 COMMON /TTRAN/ I1(5), I2(5), I3(5), I4(5), I5(5)
2 I6(5), I7(5), I8(5), I9(5), I10(5)
4 I16(5), I17(5), I18(5), I19(5), I20(5)
5 I21(5), I22(5), I23(5), I24(5), I25(5)
6 I26(5), I27(5), I28(5), I29(5), I30(5)
7 I31(5), I32(5), I33(5), I34(5), I35(5)
8 I36(5), I37(5), I38(5), I39(5), I40(5)
9 I41(5), I42(5), I43(5), I44(5), I45(5)
10 I46(5), I47(5), I48(5), I49(5), I50(5)
11 I51(5), I52(5), I53(5), I54(5), I55(5)
12 I56(5), I57(5), I58(5), I59(5), I60(5)

REAS
DIMENSION EEPRP(2,10) REA 166
DIMENSION ELNAME(3),NATOM(3) REA 167
DIMENSION EPRP(2,10),SPRP(43,30),RPRP(29,92),GPRP(124,6) REA 168
DIMENSION MP(64,16) REA 169
DIMENSION NSECTC(2) REA 170
DIMENSION SY1(3),SY2(3) REA 171
DIMENSION TNEP(186,2) REA 172
DIMENSION WCRC(4,2) REA 173
DIMENSION XRDP2(2) REA 174
EQUIVALENCE (AA(1,1),AAA(1,1)) REA 175
1 (BEBK(1),BAPD(1)) REA 176
2 (CAPX(1),DGJ(1)) REA 177
3 (SHJAT(1),SHJAP(1)) REA 178
EQUIVALENCE (CP1(1),CP2(1)) REA 179
EQUIVALENCE (DUM1(1),SUMGH) REA 180
EQUIVALENCE (ECP1(1),EEPRP(1,1)) REA 181
EQUIVALENCE (GP1(1),GPPR(1,1)) REA 182
EQUIVALENCE (NOZZLE,NPROFL(1,1)) REA 183
EQUIVALENCE (NP1(1),RPRP(1,1)) REA 184
EQUIVALENCE (SP1(1),SPRP(1,1)) REA 185
EQUIVALENCE (TNEP(1,1),TN1(1)) REA 186
EQUIVALENCE (ZP1(1),ZPKP(1,1)) REA 187
DATA ASYM /1H2,2HHE,2HLI,2HBE,1H8,1HC,1HN,1HO,1HF,2HNE,2HNA,2HMG/ REA 188
DATA SHADP /'STRA',/IGHT',/ LIN',/ E',/CIRC',/ LE B',/ OTTO',/ M',/ RE' 198
DATA SPNAME /'SP1',/ SP2',/ SP3',/ SP4',/ SP5',/ SP6',/ SP7',/ SP8'/ REA 199
DATA SY1,SY2 /31H(3,1H)/ REA 200
DATA W00D /6HTWO-DI,6MENSO,3HNL,6M AXIS,6HYMMETR,2HIC/ REA 201
DATA WORD? /3HGAP,4MDIAM/ REA 202
NAMELIST / INPUT/ REA 203
C
**** GROUP 1 - GENERAL CONTROL VARIABLES REA 204
1 IS1A,1S2A,1S3A,1S4A,1S6A,1SW1,1SW3B,TWALL,NTRAN, REA 205
1 TS0PI,CXMAX,READG,READXS REA 206
C
**** GROUP 2 - OUTPUT CONTROLS REA 207
2 IS5H,1SW7,BPRNTI,DATAF,NRECO,IRUN, REA 208
C
**** GROUP 3 - RESERVOIR CONDITIONS REA 209
1 ISW2B,PRES,FLOW,CTAPI,HSTAG,MFILTER, REA 210
C
**** GROUP 4 - GEOMETRY REA 211
4 NOZZLE,ICHAN,JDIM,NPROFL,NPROFL,NBL,DIAM,NSECTC,ISHAPE, REA 212
4 PARAM,ATPI,ZER01, REA 213
C
**** GROUP 5 - GAS MODEL REA 214
5 IGS,AAMS,ACS,JCS,OPJ,ISCI,ISST,ISRI,ICI,IE,IS,IR,ISATOM, REA 215
5 ISMOL,CTMAX,2ZER01,INEOVI, REA 216
SPRP(I,I)=SPNAME(I)
10 CONTINUE
20 CONTINUE
IF (.NOT. READ(J)) GO TO 30
READ (5,TINPUL)
WRITE (6,TINPUL)
30 CONTINUE
ICASE=ICASE+1
WRITE (6,TINPUL) ICASE,(ACOM(I),I=1,20)
LETIGIS=AAMS(AND,(IGAS.EQ.1.OR,IGAS.EQ.2)
DO 40 I=1,4
40 DU*1(I)=C
IF (ICHAN*GT*0) NPRFLS=2
IF (NPRFLS.EQ.1.OR.NPRFLS.EQ.2) GO TO 50
WRITE (6,990) NPRFLS
STOP
DO 60 I=1,NPRFLS
IF (ICHAN*GT*0) NPRCFL(I)=CP(I,ICHAN)
IPRFL=NPRCFL(I)
IF (IPRFL*GT*0) DIAM(I)=ZPRP(I,IPRFL)/1.27
60 CONTINUE
AXISYM=AXISYM
IF (NPRFLS.EQ.2) AXISYM=.FALSE.
IF (IS22<i=70.80.170
70 HS=HSTAG/1.8
WRITE (6,1000) HSTAG,FLOW
GO TO 90
80 WRITE (6,1010) PRESAI,FLOW
90 IF (NPRFS.EQ.2) GO TO 110
IF (JDIM*NE.0) GO TO 100
SMASS=70.3069*FLOW/DIAM(I)
GO TO 130
100 SMASS=HS+5173*FLOW/(DIAM(I)**2
GO TO 120
110 SMASS=70.3069*FLOW/DIAM(I)/DIAM(I)
120 IF (.NOT. SETIGS) GO TO 180
IF (IS22<i=25.80.0) GO TO 130
130 HSTAG(HSTAG
GO TO 74
140 IF (HSTAG=8000) 150,150,160
150 IGAS=2
GO TO 180
160 IGAS=1
GO TO 180
170 WRITE (6,1720) PRESAI,CTAPI
IF (.NOT. SETIGS) GO TO 180
IF (CTAPI=6000) 150.150.160
180 IF (.NOT. SETIGS) GO TO 180
IF (FSTAG*NE.0) FSTAG=SIGN(ONE,FSTAG)
PSA=1
IF (.NOT. RFLS.EQ.2) GO TO 190
NHL=1
GO TO 270
190 IF (ICHAN GT 0) NBL = CP(4+ICHAN)
200 IF (ISW29 GT 0) TSW28 = 1
LIMCUT = ISW7B*ISW36
INEQ = INEOVI
CTAP = CTAPI
PRES = PRESAI
CTMX = CTMXX
CXM = CXMAX*2.5
MBL = 1
IF (NPRFLS EQ 1) GO TO 210
IF (NHL EQ 1) MBL = 2
210 NTS = 0
DO 230 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 (NPRFLS 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 (ANGLE EQ 0 OR NRADL EQ 0) GO TO 240
IF (NXGT EQ 0 AND WXI(I) GT 1.E19) GO TO 240
IF (NPRFLS EQ 2) GO TO 240
EDGFM = TRUE.
GO TO 250
240 WEDGEM = FALSE.
250 IF (NOZLGT EQ 0) GO TO 260
XZERO = 2.54* XZERO1
GO TO 270
260 XZERO = ZPPP(2,NOZZLE)
C IN THE CASE OF A CHANNEL, THE CODE USES THE XZERO FOR THE FIRST C PROFILE
270 SL = 1.
ZZERO = ZZERO1
TSTOP = TSTOP1
TPRT = TPRTI
DELTAX = DELTIX
DELT1 = CELT1
DO 320 J = 1, NPRFL
IPRFL = NPRFL(J)
IF (IPRFL EQ 0) GO TO 300
RTMC(J) = ZPRP(I,IPRFL)
NSECTION(J) = ZPRP(3,IPRFL)
NSECTION(J) = ZPRP(4,IPRFL)
NSECTION(J) = ZPRP(5,IPRFL)
IF (J EQ 1) FACNAM = ZPRP(64,IPRFL)
NSECTION(J) = NSECTION(J)
DO 290 I = 1, NSECTION(J)
ISHAPE(I,J) = ZPRP(I+4,IPRFL)
IF (I = TNSFCTJ) ATPI(I,J) = ZPRP(I+16,IPRFL)
DO 280 L = 1.3
LL = 3*L + 24
290 CONTINUE
GO TO 320
NSEQU(J)=NSECTS(1,.J)
NSECTD(J)=NSECTS(2,.J)
NSECT(J)=NSECTU(J)+NSECTD(J)
IF (J, .EQ. 1) FACNAM=ACOM(1)
RTHCM(J)=1.27*DIAM(J)
NSECT(J)=NSECT(J)
DO 310 I=1,.NSECTJ
DO 310 L=1,3
310 PARAM(L,.J)=PARAMI(L,.I,.J)
320 CONTINUE
IF (ICHAN,.NE.0) GO TO 330
CHANAM=CP(J,.ICHAN)
FACNAM=CP(S,.ICHAN)
GO TO 340
CHANAM=ACOM(2)
340 CONTINUE
IF (NPRFLS,.EQ.1) GO TO 360
IF (ICHAN,.NE.0) GO TO 350
WRITE (6,.1030) CHANAM,FACNAM,DIAM(1),DIAM(2)
GO TO 390
CHANAM=CP(3,.ICHAN)
FACNAM=CP(5,.ICHAN)
GO TO 340
360 JDPI=JDIM+1
IF (NOZZLE,.EQ.0) GO TO 370
WRITE (6,.1050) (WORD(1,.JDPI),I=1,.3),DIAM(1),WORD2(JDPI),FACNAM
GO TO 390
370 WRITE (6,.1060) (WORD(I,.JDPI),I=1,.3),NOZZLE,DIAM(1),WORD2(JDPI),FACNAM
380 IF (NOZZLE,.EQ.0,.NOZZLE,.AND. NOZZLE,.NE.0) GO TO 440
GO TO 400
IF (ICHAN,.NE.0,.ICHAN,.AND. ICHAN,.NE.0) GO TO 440
DO 430 J=1,.NPRFLS
IF (NPRFLS,.EQ.1) GO TO 410
WRITE (6,.1070) J
410 WRITE (6,.1080) RTHCM(J),XZERO,.NSECT(J),NSECTU(J)
WRITE (6,.1090)
420 WRITE (6,.1100) K,ISH,(SHAPD(I,.ISH),I=1,.4),ATP(K,.J),(PARAM(L,K,.J)
430 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)

CONTINUE
430 CONTINUE
IF (NEELS,.EQ.0) GO TO 460
DO 450 I=1,.NEELS
IZ=EEPPL(1,.I,.IT,.I)
II=EEPPL(I,.I)
EPRLP(I,.I)=ASPML(IZ)
450 EPRLP(2,.I)=EEPPL(2,.I)
461 IGAS=IGAS
IF (IGAS<=NE*0) GO TO 470
WRITE (6, 1120)
ISC=ISCI
ISS=ISSI
ISR=ISPRI
IC=ICI
GO TO 520
470 IGAS=IABS(IGAS)
WRITE (6, 1130)'GAS=GPRP(1,IGAS)
C SET UP DESCRIPT ON OF GAS IN TERMS OF ELEMENTS, SPECIES, AND
C REACTIONS
ISC=GPRP(2,IGAS)
ISS=GPRP(3,IGAS)
ISR=GPRP(4,IGAS)
IC=GPRP(5,IGAS)
DO 480 I=1,ISC
IE(I)=GPRP(1+5,IGAS)
NCS=GPRP(120,IGAS)
DO 490 I=1,NCS
IF (IGAS.LE.0) GO TO 490
GPJ(I)=GPRP(I+IS,IGAS)
490 JCS(I)=GPRP(I+109,IGAS)
DO 500 I=1,ISS
IS(I)=GPRP(I+25,IGAS)
DO 510 I=1,ISR
IR(I)=GPRP(I+45,IGAS)
ISATOM=GPRP(121,IGAS)
ISMOL=GPRP(122,IGAS)
INT=GPRP(123,IGAS)
LEWIS=GPRP(124,IGAS)
520 IF (INT.NE.0) GO TO 530
NT=2
GO TO 540
530 NT=1
540 CONTINUE
C LOOK UP ELEMENTS
DO 550 I=1,ISC
ELEMNT(I)=EPRP(I,II)
CMW(I)=EPRP(II,II)
C LOOK UP SPECIES
IAMZIP=1
NFIT=0
DO 680 I=1,ISS
II=IS(I)
DO 660 K=1,NCS
IF (II.EQ.JCS(K)) GO TO 570
660 CONTINUE
GO TO 580
570 IJCS(K)=I
580 CONTINUE
IF (II.EQ.ISATOM) JATOM=I
IF (II.EQ.ISMOL) JMOL=I
MP(I)=SPRP(1,II)
NELS=SPRP(2,II)+0.1
DO 630 K=1,ISC
DO 590 J=1,NELS
IES=SPRP(J+2,II)+0.1
IF (IE(K),EQ,IES) GO TO 600

590 CONTINUE
LPIJ(I,K)=0
GO TO 630

600 AL=SPRP(J+5,II)
LP=AL
DAL=AL-LP
IF (ABS(DAL)*LE*0.5) GO TO 620
IF (DAL*GT*0.5) GO TO 610
LP=LP-1
GO TO 620

610 LP=LP+1
620 LPIJ(I,K)=LP
IF (I1*EQ*ISATOM.AND.*IES*EQ*1) IAMBI=2

630 CONTINUE
TFA(I)=SPRP(9,II)
TFB(I)=SPRP(10,II)
TFC(I)=SPRP(11,II)
TDD(I)=SPRP(12,II)
TDE(I)=SPRP(13,II)
TFF(I)=SPRP(14,II)
SHMP(I)=SPRP(15,II)
ETA(I)=SPRP(16,II)
SUB(I)=SPRP(17,II)
THEV(I)=SPRP(18,II)
IF (ETA(I)*LT*2.9) GO TO 650

640 THEV(I)=SPRP(K+40,II)
650 IGM(I)=SPRP(19,II)+0.1
IGJ(I)=SPRP(20,II)+0.1
IF (IGJ(I)*NE.0) NPIT=1
IF (ETA(I)*NE.0) OR(IGJ(I)*NE.0) GO TO 660
WRITE (6,1190) I,II

STOP

660 IGMV=IGM(I)
DO 670 L=1,IGMV
GELJ(L,I)=SPRP(L+20,II)
670 CONTINUE
DO 700 I=1,ISC
CAPQ(I)=0.1
DO 700 J=1,NCS
JJ=JCS(J)
NELS=SPRP(2,JJ)+0.1
DO 590 K=1,NELS
KK=SPRP(K+2,JJ)+0.1
IF (KK*NE.1) GO TO 690
CAPQ(K)=CAPQ(I)+0.1

690 CONTINUE
700 CONTINUE
C L30K UP REACTIONS

REA 495
REA 496
REA 497
REA 498
REA 499
REA 500
REA 501
REA 502
REA 503
REA 504
REA 505
REA 506
REA 507
REA 508
REA 509
REA 510
REA 511
REA 512
REA 513
REA 514
REA 515
REA 516
REA 517
REA 518
REA 519
REA 520
REA 521
REA 522
REA 523
REA 524
REA 525
REA 526
REA 527
REA 528
REA 529
REA 530
REA 531
REA 532
REA 533
REA 534
REA 535
REA 536
REA 537
REA 538
REA 539
REA 540
REA 541
REA 542
REA 543
REA 544
REA 545
REA 546
REA 547
REA 548
REA 549
DO 780 I=1,ISR
   II=IR(I)
   CAI(I)=RPR(1,II)
   ETA(I)=RPRP(2,II)
   CEACT(I)=RPRP(3,II)
   GQ(I)=RPRP(4,II)
   NRSS=RPRP(5,II)+0.1
   NPSS=RPRP(6,II)+0.1
   DO 770 J=1,ISS
      JRS=RPRP(K+6,II)+0.1
      IF (JRS.GE.IS(J)) GO TO 710
      XNUIJ(I,J)=RPRP(K+12,II)
      GO TO 720
   CONTINUE
570 XNUIJ(I,J)=0.
720 CONTINUE
   DO 730 K=1,NPSS
      JPS=RPRP(K+9,II)+0.1
      IF (JPS.GE.IS(J)) GO TO 730
570 XNUIJP(I,J)=RPRP(K+15,II)
   GO TO 740
730 CONTINUE
570 XNUIJP(I,J)=0.
740 CONTINUE
   NK=RPRP(19,II)+0.1
   DO 750 K=1,NK
      ISKUR=RPRP(K+19,II)+0.1
      IF (ISKUR.GE.IS(J)) GO TO 760
570 KUR(I,J)=0
   GO TO 770
750 CONTINUE
570 KUR(I,J)=1
770 CONTINUE
   IF (IGAS.EQ.0.OR.INT.EQ.1) GO TO 780
   KTF(I)=TNEP(I,INT)
   KTR(I)=TNEP(I+25,INT)
   ITR(I)=TNEP(I+50,INT)
   EPAR(1,I)=TNEP(I+75,INT)
   EPAR(2,I)=TNEP(I+100,INT)
780 CONTINUE
   IF (INT.EQ.1) GO TO 860
   IF (IGAS.EQ.0) GO TO 800
   DO 790 I=1,30
      TLIST(I)=TNEP(I+125,INT)
790 CONTINUE
570 PDM(I)=TNEP(I+155,INT)
   BPAR=TNEP(186,INT)
800 CONTINUE
   IF (BPAR.GE.PAR) 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
850 CONTINUE
   CONTINUE
860 CONTINUE
   DOD 850 I=1,ISR
      IF (KTF(I).NE.4) GO TO 840
DO 830 J=2,NN
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.IGAS0
IF (SUPGO) GO TO 870
WRITE (6,1140)
WRITE (6,1150)
CMW=C.
870 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*QPM(I)
IF (SUPGO) GO TO 890
WRITE (6,1160) I,SPRP(1,JJ),JJ,OPJ(I),CMW,(SY1(K),ELNAME(K),SY2(K),REA(I),NATOM(K),K=1,NELS)
890 CONTINUE
IF (*.NOT.*SUPGO) WRITE (6,1170) CMW
IGAS0=IGAS
NOZO=NOZZLE
CHAN=CHNC
IGAS=IGAS0
IF (XMODP1) 900,900,910
900 XMODP1=1.E20
WRITE (6,1180)
910 XPM1=XMODP1*2.54
IF (CMAXI*GT.*XMODP1*AND.*NMOLP*GT.1) GO TO 920
FACMP=1.*10.
GO TO 930
920 FNMP=NMOLP-1
EXPMP=1./FNMP
FACMP=(CMAXI/XMODP1)**EXPMP
930 CONTINUE
IF (NRECF+LE.00.OR.ICASE*GT.1) GO TO 950
DO 940 I=1,NRECF
940 READ (1,IPOUT)
950 CONTINUE
RETURN
C
C
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=IZEH0-ICPU
40 RETURN
END
SUBROUTINE RESTMP
C
CALCULATES RESERVOIR TEMPERATURE FROM MASS FLOW DATA.
JO
DOUBLE PRECISION AM,BM
LOGICAL ERR,SUPGO
REAL ACCM(20),EMLNT(10),HP(20)
DOUBLE PRECISION AA,AAA,CAPX,GJ,CDIJ
COMMON AA(22,24),GJ(20),CAPX(20)
COMMON A,* AFNTS, AFNK, AMACH, AR, ARBA, ARBB, BZERO,
1 C, CARR, CH, CHA, CLNT, CM, CMA, CRA, CRP,
2 CRRS, CTS, CTA, CT, CTAP, CTB, CT, CTMAX, CTMXW,
3 CTP, CTPL, CS, C, CXB, CXM, DMAX, DT, DFT, DELT1,
4 DELT2, DELTAX, DLOG, DLOGC, DT, DNT, DUT, FLUX, HDLX,
5 SCPTEST, PRES, PRESB, PRESB, PRESTH, PRHC, RHAP, RHOB,
6 RHOBAR, RHUC, RHST, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
7 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
8 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
9 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
10 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
12 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
14 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
15 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
16 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
17 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
20 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
21 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
22 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
23 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
26 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
28 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
29 RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP, RP,
N=N+1
IF (N.GT.20) GO TO 160
D0 20 I=1,2
J=3-I
CTSAVE(J+1)=CTSAVE(J)
20 F(J+1)=F(J)
CALL INTA
CALL INTA
IF (ERR) RETURN
CALL NRMAX
IF (ERR) RETURN
CTSAVE(1)=CTAP
F(1)=CMA*S*M
IF (IS=5A*NE=0) WRITE (6,RSTMP)
IF (N-2) 30,40,50
C ITERATE TO GET SECOND POINT
30 CTAP=F(1)/CONST
GO TO 10
C LINEAR INTERPOLATION TO GET THIRD POINT
40 DCT=CTSAVE(2)-CTSAVE(1)
D1=F(1)*CTSAVE(2)-F(2)*CTSAVE(1)/DCT
D2=F(2)-F(1)/DCT
CTAP=-D1/(D2-COAST)
DCTAP=ABS(CTAP-CTSAVE(1))
IF (DCTAP/ABS(CTAP)-LE.*ERR) GO TO 190
GO TO 10
C QUADRATIC INTERPOLATION
50 D0 60 I=1,3
AM(I,1)=1.0000000
AM(I,2)=CTSAVE(I)
AM(I,3)=CTSAVE(I)*CTSAVE(I)
60 BM(I)=F(I)
CALL SIMQ (AM,BM,3,KS)
IF (KS*NE=0) GO TO 40
DMC=BM(2)-CONST
IF (BM(3)*NE=0.) GO TO 70
CTAP=-BM(1)/DMC
GO TO 90
70 CTAP=-DMC+SIGN(DMC-AM-4.*BM(1)*BM(3),DMC)*0.5/BM(3)
80 DCTAP=ABS(CTAP-CTSAVE(1))
IF (DCTAP/ABS(CTAP)-LE.*ERR) GO TO 190
IF (DCTAP*GE.DCTAP) GO TO 40
DCTAP=DCTAP
GO TO 10
C ENTHALPY-MASS FLOW OPTION
90 PRESA=4.*BE-3*SMASS*SORT(HS)
MN=1+E-6*HS
SMARR=1.*E-5*SMASS
CTAP=10000.
100 N=N+1
IF (N.GT.20) GO TO 160
TLAST=CTAP
PLAST=PRESA
N2=3
110 N2=N2+1
IF (N2 GT 20) GO TO 160
H0 = H
CALL INIT
CALL INTA
IF (ERR) RETURN
H = CMA * CRP / CMA
IF (ISW5A NE 0) WRITE (6, RSTDP2)
IF (ABS(H - HS) * 1E + HERR) GO TO 130
IF (N2 GT 1) GO TO 120
CTAP = CTAP
CTAP = CTAP * HS / H
GO TO 110
CTSV = CTAP
CTAP = (HS - H) * (CTAP - CTAP) / (H0 - H)
CTAPP = CTSV
GO TO 110
N3 = 0
140 N3 = N3 + 1
IF (N3 GT 20) GO TO 160
CALL NRMAX
IF (ERR) RETURN
SMR = SMR + RAP * Sort(RGAS * CTAP / CMA)
IF (ISW5A NE 0) WRITE (6, RSTDP2)
IF (ABS(SMR - SMASS), LE + SMERR) GO TO 150
PRESA = PRESA * SMASS / SMR
CALL INTA
IF (ERR) 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, RSTDP2)
GO TO 180
170 WRITE (6, RSTDP3)
180 CALL DUMP (RNAME)
190 RETURN
200 FORMAT (26HITRACE OF RESTMP OPERATION)
210 FORMAT (30H0CONVERGENCE FAILURE IN RESTMP)
END
SUBROUTINE RNKT
LOGICAL ERR, FAILED, TEST
REAL S1, S2, S3, T1, T2, T3, Z
REAL ACOM(30), ELFMENT(10), HPL(20)
DOUBLE PRECISION AA, AAA, CAPX, GJ, CDIJ
COMMON AA(22, 24), CDIJ(20, 20), CAPX(20)
COMMON
1 A, AFN5, AFN6, AMAC, AR, ARB, ARB6, BZERO
2 C, CARB, AA, CH, CLNT, CM, CMA, CRA, CRP
3 CTP, CTPL, CTPM, CTPN, C14, C2X, C3X, C4X, C5X, C6X, CMAX, DATTEST, DTBTEST, DELT1
4 DELT2, DELTAX, DLOGA, DLOGB, DT, ENT, FLUX, HDELX, PCT
5 SPECT, PRES, PSE, PSF, PRES, PSE, PSE, PRES, PSE, PRES, PSE, PRES, PSE, PRES
6 RHOBAR, RHOC, RHOP, RHPL, RHPT, ROBARA, ROBARB, SCPG, SDE, TNT
7 SEN, SSL, SNN, SUN, SU, SSD, SUMG, TEST, TESPP, TSTPP, TSTP, UP, ZP, ZPA
8 COMMON BE(64), BET(20), BLK(31), CAT(64), Cятх(20), RNK
9 1 CCPJ(20), CEACT(24), CGI(20), CMU(20), CHI(64), CHIII(20), RNK
10 2 CLN1(64), CLNP1(64), CNW(20), ETA1(64), ETAJ(20), GJA(20), RNK
11 GJB(20), PERTGJ(20), PGJ(20), PI(64), PICH(64), RNK, RNK
12 QG(64), SAJ(20), SDCHI(64), SENT(20), SHJ(20), SHJA(20), RNK
13 5 SKL(20), SS(20), TB(20), TFA(20), TFB(20), TF(20), RNK
14 6 TFD(20), TFJ(20), TFK(20), THEV(20), XMJAT(20), XNU(64), RNK
15 COMMON BFTJ(64, 20), ELJ(10, 20), GELJ(10, 20), RNK
16 1 XNUIJ(64, 20), XNUJP(64, 20), RNK
17 COMMON IC, IM, INEQ, INEOY, IP, IRUN, ISC, ISCP1, RNK
18 1 ISPC, I5CMNR, ISP, ISS, ISSNR, ISSPI, ISSP2, ISSP3, ISSP4, RNK
19 2 ISNA, ISNA2, ISNA3, ISNA4, ISNA5, ISNA6, ISNA7, ISNA8, RNK
20 3 IS5A, IS6A, IS7A, IS8A, IUPD, IZERD, JJK, LC, RNK
21 4 NIT, NNN, NNS, NOS, NOT, NTEST, RNK
22 COMMON IGJ(20), IGK(20), ITA(5), KUR(64, 20), LPIJ(20, 10), RNK
23 COMMON ACOM, ELFMENT, HP, RNK
24 DIMENSION AA(22, 24), BTA(64, 20), CAP0(31), CCI(20), DGJ(20), GJ(20), SBRN(32), RNK
25 1 J(20), SDGJ(20), SHAP(20), THEVP(20), RNK
26 EQUIVALENCE (AA(1, 1), AAA(1, 1), BTA(1, 1), BTA(1, 1), RNK
27 1 (BLK(1), CAP0(1), (CAPXTH(20), CCI(1), (GJA(1), DGJ(1)), RNK
28 2 (CAPX(1), GJ(1)), (SAJ(1), SBJ(1), (SS(1), SDGJ(1)), RNK
29 3 (SHJA(1)), THEV(1), THEVP(1), RNK
30 COMMON /ERROR/ ERR, RNK
31 COMMON /TENER/ SDE, CETB, DCHA, CHS, SDCH, DMAX, RNK
32 COMMON /ERROR/ FAILED, RNK
33 COMMON /TENER/ FAILD, RNK
34 COMMON /TENER/ FAILD, RNK
35 COMMON /TENER/ FAILD, RNK
36 COMMON /TENER/ FAILD, RNK
37 COMMON /TENER/ FAILD, RNK
38 COMMON /TENER/ FAILD, RNK
39 COMMON /TENER/ FAILD, RNK
40 COMMON /TENER/ FAILD, RNK
41 COMMON /TENER/ FAILD, RNK
42 COMMON /TENER/ FAILD, RNK
43 COMMON /TENER/ FAILD, RNK
44 COMMON /TENER/ FAILD, RNK
45 COMMON /TENER/ FAILD, RNK
46 COMMON /TENER/ FAILD, RNK
47 COMMON /TENER/ FAILD, RNK
48 COMMON /TENER/ FAILD, RNK
49 COMMON /TENER/ FAILD, RNK
50 COMMON /TENER/ FAILD, RNK
51 COMMON /TENER/ FAILD, RNK
52 COMMON /TENER/ FAILD, RNK
53 COMMON /TENER/ FAILD, RNK
54 COMMON /TENER/ FAILD, RNK
55 COMMON /TENER/ FAILD, RNK
IFAIL=0
TE=NT*EQ.1
LIM=LSSP1
IF (iNOT TE) LIM=LIM+2
GPRE=GPRE
DQ=MAX
IF (I$EQ 0) GO TO 10
IST.NT=1
WRITE (6,410) ISTMNT
WRITE (6, RNKOMP)
10 DO 20 J=1,ISS
F1(J)=DGJ(J)
GJ1(J)=GJP(J)
CT=CTB+DT*HDELX
GJ2(ISSP1)=CT
IF (TE) GO TO 30
F1(ISSP2)=DTE
GJ1(ISSP2)=CTEB
GJ2(ISSP2)=CT
F1(ISSP3)=CHA
GJ1(ISSP3)=CHA
GH=CH3+CHA*HDELX
GJ2(ISSP3)=CHA
30 DO 40 J=1,LIM
IFAIL=-J
40 IF (GJ (J) .GE. 270.40.40) CONTINUE
IFAIL=0
CXX=CXX+HDELX
CALL DERIVS
IF (ERR*NR*FAILED) RETURN
IF (I$EQ 0) GO TO 50
ISTMNT=2
WRITE (6, 410) ISTMNT
WRITE (6, RNKOMP)
50 DO 60 J=1,ISS
F2(J)=DGJ(J)
GJ2(J)=GJ(J)*DGJ(J)*HDELX
60 DO 82 J=1,ISS
F2(ISSP1)=DTE
GJ2(ISSP1)=DTE
GJ2(ISSP1)=CT
IF (TE) GO TO 70
F2(ISSP2)=DTE
GJ2(ISSP2)=DTE
GJ2(ISSP2)=DTE
GJ2(ISSP2)=CT
GJ2(ISSP3)=CHA
GJ2(ISSP3)=CHA
70 DO 82 J=1,LIM
IFAIL=-J
IF (GJ3(J)) 270,80,80
CONTINUE
IF (TE) GO TO 90
IF (FAIL=1*SSP4)
IF (ABS(DPPE-OCPE8)*LE.DQRK.ORB.DQMAX*EQ.0.0) GO TO 90
GO TO 270
90 IFAIL=0
CALL DERIVS
IF (ERR.OE*FAILEC) RETURN
IF (ISW59.EQ.0) GO TO 100
ISTMNT=3
WRITE (6,410) ISTMNT
WRITE (6,RNKDMP)
D3 110 J=1,ISS
110 F3(J)=DGJ(J)
F3(ISSP1)=DT
IF (TE) GO TO 120
F3(ISSP2)=DTE
F3(ISSP3)=DCHA
DO 150 J=1,1LM
IF (F1(J)EQ.0.0) GO TO 130
X4(J)=F2(J)=F1(J)
130 P(J)=0.0
GO TO 150
140 XX=ABS(F3(J)-F2(J))
IF (XX*EO.5*EQ.0.0) GO TO 130
XX=ALC1G1XX
YY=ALC1G1DAPIS(X4,J))
IF ((XX-YY)*GT.3E+0) GO TO 130
P(J)=2.5*(F3(J)-F2(J))/X4(J)
150 CONTINUE
D3 210 J=1,1LM
IF (P(J)) 160,210,210
160 IF (P(J)+1.25) 260,200,170
170 Z=3.0
S1=0.0
S2=0.0
S3=0.0
T1=1.0
T2=0.5
T3=1.0/6.0
180 S1=S1+T1
S2=S2+T2
S3=S3+T3
Z=Z+1.0
T1=T1*P(J)/(Z-2.0)
T2=T2*P(J)/(Z-1.0)
T3=T3*P(J)/Z
IF ((S1+F1)-S3) 180,190,180
190 X1(J)=S1
X2(J)=S2
X3(J)=S3
GO TO 210
200 X1(J)={EPY(P(J)-1.0)/P(J)}
<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>210</td>
<td>G 1</td>
<td>BEGINNING</td>
</tr>
<tr>
<td>220</td>
<td>G 2</td>
<td>FUNCTION</td>
</tr>
<tr>
<td>230</td>
<td>G 3</td>
<td>FOR N = 0 TO 250</td>
</tr>
</tbody>
</table>
IF (TE) GO TO 390
SDTE=SDQ(ISSP2)
SDCHA=SDQ(ISSP3)
390 IF (IS*58.EQ.0) GO TO 400
ISTMNT=5
WRITE (6,410) ISTMNT
WRITE (6,PKDMP)
WRITE (6,PKDP2)
400 RETURN

FORMAT (18H DUMP AT STATEMENT,12)
END
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 (RADIANS)  
FL(2) = MACH NUMBER BEHIND SHOCK  
FL(3) = DENSITY RATIO, RHO(BEHIND)/RHO(INFINITY)  
FL(4) = STATIC PRESSURE RATIO, PB(behind)/PB(INFINITY)  
FL(5) = STATIC TEMPERATURE RATIO, TB(behind)/TB(INFINITY)  
FL(6) = TOTAL PRESSURE RATIO, PT(behind)/PT(INFINITY)  
FL(7) = TSTAG/TINF

IEIR=1 NORMAL RETURN, IERR=2 ERROR RETURN, IERR=0 DETACHED SHOCK  

DIMENSION FL(7)  
KIT=0  
DEL=DEL/57.*295779  
SD=SIN(DELR)  
CD=COS(DELR)  
BM2=BM*F**2  
DM4=BM*F**2  
C1=GAM+1.0  
C2=GAM+1.0  
B2=(3*M2+2*0)/BM2=GAM*SD**2  
C=(2.*M2**2+1.)/BM4+(C1**2/4.*C2/RM2)*SD**2  
D0=C0**2/RM4  
C3=2.*B  
UG=0  
FU=UG*(C+UG*(B+UG))  
FP=UG*(C3+UG)  
UN=UG-FU/FP  
IF (OABS(UG/UN-1.*DC)-1.0<8) 50 50 20  
IF (KIT=50) 30 30 40  
KIT=KIT+1  
UG=UN  
GO TO 10  
WRITE (6,100) IERR=2  
RETURN  
50  
ALP=U+UN  
BETA=UN*ALP+C  
DIS=ALP**2-4.*BETA  
IF (DIS*GE0) GO TO 60  
WRITE (6,110) IERR=2  
RETURN  
60  
DIS=SGRT(DIS)  
UI=(-ALP+DIS)/2.0  
U2=(-ALP-DIS)/2.0  
IF (U1-U2) 70 70 80  
RT=UI  

SHK 1  
SHK 2  
SHK 3  
SHK 4  
SHK 5  
SHK 6  
SHK 7  
SHK 8  
SHK 9  
SHK 10  
SHK 11  
SHK 12  
SHK 13  
SHK 14  
SHK 15  
SHK 16  
SHK 17  
SHK 18  
SHK 19  
SHK 20  
SHK 21  
SHK 22  
SHK 23  
SHK 24  
SHK 25  
SHK 26  
SHK 27  
SHK 28  
SHK 29  
SHK 30  
SHK 31  
SHK 32  
SHK 33  
SHK 34  
SHK 35  
SHK 36  
SHK 37  
SHK 38  
SHK 39  
SHK 40  
SHK 41  
SHK 42  
SHK 43  
SHK 44  
SHK 45  
SHK 46  
SHK 47  
SHK 48  
SHK 49  
SHK 50  
SHK 51  
SHK 52  
SHK 53  
SHK 54  
SHK 55
GO TO 90
RT=U2
80 ZT= SQRT(RT)
90 CONTINUE
C FL(1)=ATAN(ZT/SQRT(1-ZT**2))
V1=BM2*RT
C V2=C1**2*BM4*RT-4.0*(V1-1.0)*(GAM*V1+1.0)
C V3=(2.0*GAM*V1-C2)*(C2*V1+2.0)
C FL(2)= SQRT(V2/V3)
C FL(3)=C1*V1/(C2*V1+2.0)
C FL(5)=V3/C1**2/V1
C FL(4)=(2.0*GAM*V1-C2)/C1
C C4=C2/2.0
C BM2=FL(2)**2
C CS=GAM/C2
C SMFP=1.0+C4*BM2
C SMFP=1.0+C4*BM2
C FL(6)=(SMFP/SMFP)**C5*FL(4)
C FL(71)=SMFP
RETURN
100 FORMAT (32H EXCEEDED 50 ITERATIONS IN SHOCK)
110 FORMAT (110H ----SHOCK DETACHED IN CLASSICAL WEDGE CALCULATION. UN
1MODIFIED 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 (GINV) AND MATRIX PRODUCT (GMPRO).
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
ELE%MTS.
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. THIS
TOLERANCE CAN BE MODIFIED BY REPLACING THE FIRST STATEMENT.

SUBROUTINE SIMO(A,B,N,KS)
KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC VERSION
DIMENSION A(N),B(N)
DATA TOL/0.001/
KS=0
IF (N.GT.1) GO TO 10
SPECIAL CASE N=1
IF (DABS(A(1)) .LE. TOL) GO TO 40
B(1) = B(1)/A(1)
RETURN
FORWARD SOLUTION
10 JJ = -N
DO 90 J = 1, N
JJ = JJ + N + 1
BIGA = 0
IT = JJ - J
DO 30 I = J, N
SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN
20 IF (DABS(BIGA) - DABS(A(IJ))) .GE. 20, 30, 30
BIGA = A(IJ)
IMAX = I
CONTINUE
TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)
40 IF (DABS(BIGA) - TOL) .LT. 40, 40, 50
KS = 1
INTERCHANGE ROWS IF NECESSARY
50 I1 = J + N*(J - 2)
IT = IMAX - J
DO 60 K = J, N
I2 = I1 + IT
SAVE = A(I1)
A(I1) = A(I2)
A(I2) = SAVE
DIVIDE EQUATION BY LEADING COEFFICIENT
60 A(I1) = A(I1)/BIGA
SAVE = B(IMAX)
B(IMAX) = J(J)
B(J) = SAVE/BIGA
ELIMINATE NEXT VARIABLE
70 IF (J - N) .LE. 70, 100, 70
IQS = N*(J - 1)
DO 90 IX = JY, N
IX = IQS + IX
IT = J - IX
DO 80 JX=JY+N
    IXJX=N*(JX-1)+IX
    JJX=IXJX+1T
  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=1
    DO 110 K=1, J
    B(IB)=B(IB)-A(IA)*B(IC)
    IA=IA-N
  110 IC=IC-1
RETURN
END
LIM1=2
CTMXX=-1000.
DO 40 K=1,4
DO 30 I=1,74
30 TPROP(I,K,1)=0.
40 CONTINUE
DO 50 M=LIM1*LIM
'T=0
II=1
TV=0.
DO 70 I=1,74
IT=IT+1
IZ=1
IF (II.LE.NDT(II)) GO TO 60
II=II+1
IT=1
60 TV=TV+DELT(I)
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))*TPR/1000.
TF=P(I,3,M)=CGPJ(J)*CRA
T-I=(I,4,M)=SFNT(J)*CRA
IF (CT-1.*70.50.*80.
70 CONTINUE
80 CTMXX=-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 STUNI2
WRITE (6,436)
WRITE (6,340)
WRITE (6,350)
DO 150 L=1,NNKO
I=ISEQ(L)
NNQI=NNQ(I)
DO 140 J=1,NNQI
IIQJ=IIQ(J)
JJQJ=JJQ(J)
140 WRITE (6,360) L,I,KKO(I),(VV(K,I),K=1,5),SPRP(1,IIQJ),SPRP(1,JJQJ)
DO 120 J=1,120,130
120 WRITE (6,360) L,I,KKO(I),(VV(K,I),K=1,5),SPRP(1,IIQJ),SPRP(1,JJQJ)
GO TO 140
130 WRITE (6,370) SPRP(1,IIQJ),SPRP(1,JJQJ)
140 CONTINUE
WRITE (6,330)
WRITE (6,380)
WRITE (6,350)
MV2=1
LQ2=0
DO 190 L=1,NNKO
KK=KKQ(L)
190 WRITE (6,330)
MV1=MV2+1
MV2=MV2+NV(KK)
L01=L02+1
L02=N0(L)
WRITE (6,390) L,KK,(V(I),I=MV1,MV2)
IF (L02.LT.L01) GO TO 190
DO 150 L0=L01+1,L02,1
I=10(L0)
J=J0(L0)
160 IF (L0.LT.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 210 I=1,ISS
SAVEC(I)=GJA(I)*CMA
200 TV=1000*K
IF (TV.GT.1.0001*CTAP) GO TO 220
ICARD=ICARD+1
CALL T2ANSP (TV,PRESA)
WRITE (6,270) TV,(G(L,J),L=1,3),J,HP(I),HP(J),ICARD
IF (ISW1B.GE.-1) GO TO 210
PUNCH 270, TV,(G(L,J),L=1,3),J,HP(I),HP(J),ICARD
210 CONTINUE
220 CONTINUE
230 CONTINUE
WRITE (6,280)
240 RETURN

C

250 FORMAT (1H1,38X,56HAVERAGED PAIR CROSS-SECTIONS AS FUNCTIONS OF TEST
14PRESSURE)
260 FORMAT (14/2X,55TEMP.,6X,4HQ(1),8X,4HQ(2),8X,4HQ(3),5X,7HINDICES,5STU
1X,7HSPECIES,8X,5COUNT)
270 FORMAT (F7.0,IP3E12,3,5X,13-I=1,3,3X,4,14-I=A4,SX,18)
280 FORMAT (F7.0,IP3E12,3,5X,13-I=1,3,3X,4,14-I=A4,SX,18)
15ATUERES ARE BASED ON ELECTRON MOLE FRACTION IN RESERVOIR)
290 FORMAT (1H1,40X,22HTHERMAL PROPERTIES OF ,A4,6H (H00=+F10.3,11H KCS
14/4/MOLE))
300 FORMAT (1H1,10X,3A6,18PHYSICAL MODEL**,3A6/1H0,8X,1HT,2X,12H (MUSTU
10-H00)/RT,9X,5H-H00,12X,2HCP,12X,2H50/29X,9HKCAL/MOLE,2X,12HCAL/STU
20LE-DEG,2X,12HCAL/MOLE-DEG)
310 FORMAT (1H3,11X,3A6,18PHYSICAL MODEL**,3A6/1H0,8X,1HT,2X,12H(MU0-H00)/RT,9X,5H-H00,12X,2HCP,12STU
12X,2H50/12X,12H(MU0-H00)/RT,9X,5H-H00,12X,2HCP,12STU
2X,2H50/29X,9HKCAL/MOLE,2X,12HCAL/STU
3AC/DEG,2X,12HCAL/MOLE-DEG,2X,12HCAL/MOLE-DEG,29X,9HKCAL/MOLE,2X,12HCAL/STU
SUBROUTINE THERM

SPECIES THERMAL PROPERTIES

LOGICAL MIX, F1

REAL ACOM(30), ELMENT(10), MR(20)

DOUBLE PRECISION AAAAA, CAPX, GJ, CDI

COMMON AA(22, 24), CDI(20, 10), CAPX(20)

COMMON A, AFNTS, AFNX, AMACH, AR, ARBA, ARBB, BZERO

1

C, CAR, CHA, CLNT, CM, CK, CRA, CRP

3

CTP, CTPL, CTT, CX, CXB, CMAX, DATEST, DBTEST, DELT1

4

DELT2, DELTAX, DLOGA, DT, EN, FLUX, HDELX, PCT

5

SPT, PRE, PREB, PREST, RP, RHAP, RHAP, RHOB

6

CMOC, RHO, ROBAR, ROBAR, SCG, SOT

7

SEN, SHG, SC, SL, SL6, SM, SU, SU2, SUMG

8

TEST, TESTA, TPRINT, TSTOP, UP, ZP, ZPA

COMMON RE(64), RET(20), BLBK(31), CA(64), CAPXTH(20)

3

GJ(20), CEACT(64), CMU(23), CHI(64), CHIL(20)

2

CMOC, RHO, RP, RHO, RTMAX, SORBAR, SORBAR, SCG, SOT

3

SEN, SHG, SC, SL, SL6, SM, SU, SU2, SUMG

4

TEST, TESTA, TPRINT, TSTOP, UP, ZP, ZPA

COMMON IC, TM, INEQ, IOV, IP, IRUN, ISC, ISCP, TP

1

ISMC, ISMCNR, ISR, ISS, ISSN, ISPP1, ISPP2, ISSP3, ISSP4

2

SAX1A, ISAB, ISA1B, ISW2A, ISW2B, ISW3A, ISW4B, ISW5A

3

ISW6A, ISW6B, IUPD, JJK, LBK, M1, NFIT

4

NIT, NNN, NNS, NQ, NQ, NTEN

COMMON TGG(30), IGM(20), ITB(5), KUR(64, 20), LPIJ(20, 10)

2

COMMON ACOM, ELMENT, HP

DIMENSION AAA(22, 24), BTA(64, 20), CAPX(31), CC(20), DGJ(20), GJ(20), SBTH(20)

3

EQUIVALENCE (AA(1, 1), AAA(1, 1), (BETA(1, 1), BTA(1, 1)

1

(CLA(1), GJ(1), (SA(1), SB(1)), (SS(1), SDGJ(1)))

2

3

COMMON /POLYAT/ THEVE(420)

C

COMPUTE ALL THERMAL PROPERTIES

IF*FALSE*

GO TO 10

ENTRY THERM

C

COMPUTE SPECIFIC HEATS ONLY

IF*TRUE*

10

CLG = ALOG(CT)

CPT = CT*CTAP

CLN = ALOG(CTP)

G = 320, J = 1, ISS

Z1 = 0

Z2 = 0

IF (ETAJ(J, LE, 0)) GO TO 30

IF (TGJ(J) > 20, 30, 20)

ZPT = 1, G = 500

IF (CTMX, LE, 10) GO TO 40

IF (CTMX, LE, 10) GO TO 40
MIX=TRUE
MMF=(T + 500)
GOT050
30 IF (ETA(J)) 50,50,70
40 MIX=FALSE
50 IF (E1) GO TO 60
C
60 XAI(J)=FAB(J)*CI*TFK(J)*CT*TFC(J)*2.0*TFD(J)
THE 68
1/J/3.0+CT*TFE(J)/4.0)+SHJ(J)/CT
THE 69
SHJ(J)=SHJ(J)+CT*TFA(J)+CT*TFC(J)+CT*(TFD(J)+CT*TFE(J))
THE 70
1/J)
THE 71
60 CCP(J)=FAB(J)+CT*(1.0+TFB(J)+CT*(3.0+TFC(J)+CT*(4.0+TFD(J)+CT*(5.0)
THE 72
THE 73
IF (NDT=MIX) GC TO 300
THE 74
XMTF=XMJAT(J)
THE 75
SHTF=SHJ(J)
THE 76
CCPTF=CCP(J)
THE 77
STF=SHTF/CT-XMTF
THE 78
C
70 CCP(J)=0
THE 79
IF (E1) GO TO 80
THE 80
SHJ(J)=0
THE 81
XMJAT(J)=0
THE 82
80 IF (ETA(J)>LT.3) GO TO 150
THE 83
C
80 CCP(J)=0
THE 84
IF (E1) GO TO 60
THE 85
C
80 CCP(J)=0
THE 86
IF (E1) GO TO 60
THE 87
C
80 CCP(J)=0
THE 88
IF (E1) GO TO 60
THE 89
C
80 CCP(J)=0
THE 90
IF (E1) GO TO 60
THE 91
C
THE 92
THE 93
THE 94
THE 95
THE 96
THE 97
THE 98
THE 99
THE 100
THE 101
THE 102
THE 103
THE 104
THE 105
THE 106
THE 107
THE 108
THE 109
THE 110
CONTINUE
DRIGWAL:

C TRANSLATIONAL, ROTATIONAL, AND VIBRATIONAL CONTRIBUTIONS FOR
C MONONOMIC AND DIATOMIC SPECIES

150 Z1=ETAJ(J)-1.0
Z2=Z1+2.5

160 CCPJ(J)=0.

170 Z3=THEV(J)

180 Z3=THEV(J)/CT

190 Z4=EXP(Z3)
Z5=Z4-1.

200 Z5=THEV(J)/Z5

210 Z7=Z5/25

220 Z3=ALOG(Z4/Z5)
Z9=Z7*(Z4*77)

230 CCPJ(J)=Z1*Z9

240 CCPJ(J)=CCPJ(J)+Z2

250 IF (F1) GO TO 249

260 IF (SAJ(J)>1.D-37) GO TO 270

270 xx=52.5

THE 111
THE 112
THE 113
THE 114
THE 115
THE 116
THE 117
THE 118
THE 119
THE 120
THE 121
THE 122
THE 123
THE 124
THE 125
THE 126
THE 127
THE 128
THE 129
THE 130
THE 131
THE 132
THE 133
THE 134
THE 135
THE 136
THE 137
THE 138
THE 139
THE 140
THE 141
THE 142
THE 143
THE 144
THE 145
THE 146
THE 147
THE 148
THE 149
THE 150
THE 151
THE 152
THE 153
THE 154
THE 155
THE 156
THE 157
THE 158
THE 159
THE 160
THE 161
THE 162
THE 163
THE 164
THE 165
SUBROUTINE TRANSX (ITAB,P)
SUBROUTINE TRANSX HAS GENERAL SUPERVISION OF ALL TRANSPORT

PROPERTY CALCULATIONS IN NATURAL CODE.

LOGICAL ERR
REAL AC(10),ELMNT(10),HP(20)
DOUBLE PRECISION AA,CAPX,CGI
COMMON AA(22,24),CGI(22,10),CAPX(20)
COMMON AZ,AFNTY,AFN,ANACH,AR,ARRA,ARRB,BZERO,
1 CZ,CARR,CH,CHA,CLNT,CM,CMA,CRA,CRP,
2 CPRR,CRS,CSTA,CT,CTAP,CTB,CTC,CTMXX,CTMXK,
3 CTPL,CTT,CRX,CRZ,CMAX,DATE,DATE,DELT,
4 DELT,DETX,DLG,DLOG,DLGS,DT,DTT,FLUX,HOELX,PCT,
5 SPECT,SRES,PREA,PREA,PREB,PREAC,RPA,RHA,RHO,
6 RHCBAR,RHOC,RHOP,RHPL,RHTR,ROBARA,ROBARB,SCPQ,SDT,
7 SEN,SHPR,SC,SL,SL64,SM,SM,SU,SU2,SUMG,
8 TEST,TESTB,PRINT,TSTOP,UP,VA,ZPA

COMMON BB(2,4),BS(2),BLB(31),CAB(64),CAPXTH(20),TRA
1 CCPJ(20),CFACT(64),CG(20),CMU(20),CHI(64),CHII(20),TRA
2 CNNC(64),CLNPI(64),CM(22),ETA(64),ETAJ(20),TRA
3 GJ(20),PGJ(20),P(64),PICH(64),QM(20),TRA
4 GJ(64),SAJ(20),SDCH(64),SEN(20),SHJ(20),TRA
5 SKL(20),SS(20),TB(20),TRA
6 TFD(20),TFE(20),TFK(20),THEV(20),XMFAT(20),XNU(64),TRA
7 COMC BETA(64,20),ELJ(10,20),FLJ(10,20),TRA
1 XNUJ(64,20),XNUJX(64,20),XNUJIP(64,20),TRA

COMMON IC,ICG,ICGQ,IGEQ,IP,IRUN,ISCC,ISC,IP,
1 ISC,ISCCM,IGA,ISS,ISSA,ISSB,ISSC,ISSD,ISSP4,TRA
2 ISWA,ISWA,ISWA,ISWB,ISWA,ISWC,ISWD,ISWE,ISWE,TRA
3 ISWA,ISWA,ISWA,UPD,IZEF,JKK,LC,M1,NFIT,
4 AIT,ANN,ANN,ANN,ANN,ANN,TRA

COMMON IGJ(30),IGM(20),ITB(6),KUR(64,20),LPIJ(20,10),TRA
1 COMMON ACON,ELMNT,HP
2 COMMON /TRANS/ VISC,PRF,SRMA,FLFWI
3 COMMON /TRANS/ T,(3,20),TM(20),TRA
4 COMMON /TRANS/ T(20,20),TM(20,20),TRA
5 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
6 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
7 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
8 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
9 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
10 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
11 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA
12 COMMON /TRANS/ T,(3,20,20),TM(20,20),TRA

C ENTRY TRANSX OF TRANSX DEFINES SPECIES AND INITIALIZES DATA FOR
C TRANSPORT CALCULATIONS.
C
C @ISS IS THE NUMBER OF SPECIES TO BE USED IN THE TRANSPORT
C
C CALCULATIONS:
C
C CALL XSECT
C IF (ERR) RETURN
C CGI IS THE SPECIES MOLECULAR WEIGHT.
DO 10 IC=1,N
10 Z(I)=2.0*CGI(I)
DO 20 IC=1,N
DO 20 J=1,N
Q(1,J)=ZM1(I)+ZM1(J)
C(J,I)=(ZM1(I)-ZM1(J))/Q(1,I,J)
RETURN
C
ENTRY TRANSF CALCULATES TRANSFER PROPERTIES FOR TEMPERATURE T.
C
PRESSURE, P, AND MOL FRACTIONS X(J),
C
NOTE--THE MOL FRACTIONS ARE NOT NECESSARILY IN EQUILIBRIUM AT
C
THE TEMPERATURE T.
T=TTAB
C
SAVE CT AND CCP.'(J) FOR LATER USE.
CTSAVE=CT
CT=CT/CTAB
DO 40 J=1,N
CTSAVE(J)=CCP(J)
C
CALCULATE SPECIES SPECIFIC HEATS AT TEMPERATURE T.
CALL THERM
CPTCT=3.
CMSAVE=3.
DO 50 J=1,N
X(J)=SAVEC(J)
DH(J)=CCP(J)-2.*X(J)*CPTCT
CMSAVE=CMC(J)*X(J)+CMSAVE
CALL PUTGIN (X(I)*P)
C
COMPUTE COLLISION CROSS SECTIONS. STORE IN Q ARRAY.
CALL PUTGIN (X(I)*P)
C
COMPUTE MATRIX ELEMENTS FOR TRANSPORT CALCULATIONS. STORE IN Q
C
ARRAY.
IF (IS=18) LT 0) RETURN
DO 60 I=1,N
DO 60 J=1,N
Q(I,J)=C(I,J)*Q(2,1,J)
S=SGRT(T)
SIG=1.3451E5/S
FLEX1=CMPSD(Q(1,JD1,ID2))
IF (N.EQ.1) GO TO 80
DO 70 J=2,N
Q(3,J,J)=Q(2,J,J)
J1=J-1
DO 70 J=1,J1
Q(3,J,J)=3.*375*Q(1,J,J)-75*E1,J)*Q(3,J,J)
DO 79 J=1,J1
Q(1,J,J)=(1-C(J,J)**2)*(Q(3,J,J)-Q(2,J,J))
Q(2,J,J)=C(J,J)*Q(3,J,J)-Q(1,J,J)
Q(3,J,J)=Q(3,J,J)-Q(2,J,J)
Q(2,J,J)=Q(2,J,J)-Q(1,J,J)/Q(3,J,J)
Q(3,J,J)=Q(3,J,J)
79 DO 70 J=2,N
Q(3,J,J)=Q(3,J,J)
C
COMPUTE TRANSPORT PROPERTIES. VISC=VISCOSITY, PRF=FROZEN PRANDTL
C
NUMCP=SIGMA=ELCTRICAL CONDUCTIVITY. FLEWIS=ATOM-MOLECULE LEWIS
C
DO 90 I=1,N
Q(I,1)=Q(I,1)*Q(1,I,I)
N(1,1)=I*Q(1,I,1)
B(I,2)=X(I)*Q(2,I,I)
CALL ELCOND (SIGMA, SSIG)
CALL KINT (ZKINT)
CALL KANDMU (ZK)
ZKINT=ZK(I)+ZKINT
VISCM=5*ZK(I)*1.0*C-3
FLEWIS=FLEWIS/ZKINT*IA(I)
PRF=5*3167*CPTOT*ZK(2)/ZKINT/CMSAVE
IF (ISW8B.EQ.0) RETURN
C
PRINT TRANSPORT PROPERTY VALUES ONLY IF ISW8B.NE.0
WRITE (6,100)
WRITE (6,110) T,P,PRF,SIGMA,FLEWIS,CPTOT,CMSAVE
RETURN
C
100 FORMAT (1H )
110 FORMAT (1P12E11.3)
END

TRA 111
TRA 112
TRA 113
TRA 114
TRA 115
TRA 116
TRA 117
TRA 118
TRA 119
TRA 120
TRA 121
TRA 122
TRA 123
TRA 124
TRA 125
TRA 126
TRA 127
TRA 128
TRA 129
TRA 130-
AM = 53.3*CMAM2/PEPF
XS = 12.6*(EPS - TAU - AM)*AM**2/REP
WRITE (A, 260) UNITH
WRITE (A, 270) XM
IF (ITH = 0 * 2) WRITE (6, 280) XS
ITH = ITH
GFAC = 1.0 - GAMMA
GFAC = 1.0 - GAMMA
DO 240 IP = 1, NADLE
TN = 2*TNADLE + IR
IF (TN = 0.0) TN = 1.0 - E6
EN = AV * SORT(C/(REP*TN/12))
TAUN = TAUN
TAUN = TAUN + TAUN
TAUN = TAUN + TAUN
TAUN = TAUN + TAUN
CHD = CHD*GFAC + TAUN*EN*TN
CSK = CSK + GFAC*TAN/TAUN4
CSK = CSK + GFAC*TAN/TAUN4
YSFAC = YSFAC + GFAC*TN/TAUN4
GFAC = GFAC + GFAC(1)/TAUN2
GFAC = GFAC + GFAC(2)/TAUN2
PFAK = PFAK + GFAC
OMEGA = 1.0/PFAK
250 IAA = IANGLE
IF (IR = 0) GO TO 30
AND = AM
AND = AMAX1(ANGLE, ANMIN)
GAMMA = GAMMA
CALL SHOCK (AM, AND, GAMMA, FL, IERR)
IF (IERR = NE*0) GO TO 20
IAMAX = IA
GO TO 30
20 APR = FL*(4)
IP = IF(SR*EO*1) GO TO 30
CALL DUMP (RNAME)
RETURN
IF (IAT = IAMAX) ITH = 2
FINAL = FALSE
ALPHA = ANGLE(IA)/57.2928
FMC = 0.1
IF (ITH = 0) GO TO 40
CAPGAMMA = GFAC(1)*ALPHA
GO TO 50
IF (IAT = 0) GO TO 50
CAPGAMMA = GFAC(1)*SORT(GAMMA/A*AMAX1(APR*IA)-1*, ZERO)
GO TO 50
SSA = SIN(A*AMAX1(ALPHA, ALMIN))
SSA = SSA**2
EXPV = EXP(-SSASQ)
ERFV = 1.0 + PPF(SSA)
GFMC2(1) = GFMC2*(EXPV + SQRTI*SSA*ERFV) - 0.5*EXPV
I = 1
+1 = 1
NW2=1
ITAB=1
IT=IT+1
IF (NA1*GT*NWX) GO TO 70
WVX1=2*X1+(NW1-1)*DWX
GO TO 70
70
WVX1=1+D30
60 IF (NA2*GT*20) GO TO 90
%VX2*AX1(NW2)
GO TO 120
90 %VX2=1+D30
100 IF (%VX1-%VX2) 110,110,120
110 IF (%VX1*GT*1.E19) GO TO 130
WV(1)=WVX1
NW1=NA1+1
GO TO 140
120 IF (%VX2*GT*1.E19) GO TO 130
WV(1)=WVX2
NW1=NA2+1
GO TO 140
130 IF (;ITLE*1) GO TO 230
IT=IT-1
IVAL=.TRUE.
GO TO 120
140 XE=WV(1)*COS(ALPHA)
ZETA=ZFT*XIC
ZETA(1)=ZETA
CALL WEDSLN (ZETA,CAPGAM,ZZP,ZZPP,Z)
ZZPS=ZZP
IF (:;TLE*1) GO TO 150
C
CHENG=KLYP THEORY
YES(IT)=+YSAK*2
P(1)=INF*PFAK*ZZPP
SQT=SQIT*ZZPS
DELSTK(IT)=DFAK*SQ1/ZZPP
GO TO 150
C
VODIFKFD CHENG-KEMP THEORY
SQT=SQIT*ZZPS*OMEGA*7TA(IT))
DELSTK(IT)=DFAK*SQT/(ZZPP+OMEGA)
D4ITC=PFAK*ZZP+1
YES(IT)=-YSAK*ZTIC
YES(IT)=YSAK*(1+SQ1)/((ZZPP*OMEGA)+A*XC*TAN(ALPHA))
CH(1)=CHO/DELSTW(IT)
YES(IT)=CH(1)*PHINF*UINF*(HO-HW)*1.E8
ASTQ(1)=AST(1)
IF (:ITLE*1) GO TO 170
IF (:IT1*,L*,1*) GFMI(IA)) GO TO 170
IFM=1
GSM(IT)=GFMI(IA)
ASTG(IT)=AST(2)
GO TO 170
170 IF (:ITLE*1) GO TO 60
180 IF (:ITAB*EQ*1) GO TO 190
WRITE (1*,260)
GO TO 22
190 CPVY=CAPGAM
SUBROUTINE $\text{SOLN (ZETA,CAPGAM,ZZP,ZZP; )}$

SOLUTION OF THE CHENG EQUATION FOR FLOW OVER A WEDGE

IMPLICIT REAL*4 (A-H,O-Z)

C
KEEP THIS IMPLICIT STATEMENT IN THE UNIVAC 1103 VERSION

LOGICAL CALLED

DATA CALLED $/\text{FALSE}$/

DA $= (0.31118712.6900+1.73205980) /$

NAMELIST $/\text{FMP/ XL=SOTL.Q1.Q2.Q3.C.05.ZT. DLDZT /}$

IF (CALLED) GO TO 10

CALLED = TRUE

C1 = 1.0

C3 = 1.0/3.0

C29 = 2.0/9.0

C9 = 1.0

C63 = 6.0/3.0

C103 = 10.0/3.0

C4 = 1.0

C32 = 1.500

C2 = 0.0

C76 = 7.0/3.0

D1 = ZETA*C3

D2 = 0.0*SQRT(ZETA)

IF (ZETA > 1.0-6) GO TO 20

C
HOGE'S ANALYTICAL SOLUTION FOR SMALL ZETA

C12 = 1.0

Z = 1.0636712 + C506931 + P1 + P2 = 0.0240*ZETA

XL = 1.8171131 + C1 + 297560 + C2 + 147120 + P12

DLDZT = 0.685712 + C6298 + P2 + C92808

GO TO 30

C
ITERATIVE SOLUTION BASED ON CHENG/KEMP EXACT SOLUTION FOR GAMMA=0.

XL = 0.0 + P1 + P2

GO

IF (N GT 20) GO TO 40

SOTL = SQRT(XL)

Q1 = C1 + SOTL

Q2 = Q1 + Q1

Q3 = Q1 + Q3

Q4 = Q2 + Q2

Q5 = LOG(Q1)

ZT = C3 + 0.220 + C9 + Q2 + C63 + Q1 + 05*(C103 + C4*SOTL + C2 + Q5) + C769

Z = C2*(SOTL - XL/C2 + C3*XL**C32 - Q5)

DZT = C1/Z

IF (QAMS(ZT-ZETA)/ZETA LE 1.0-4) GO TO 50

XL = XL - (ZT-ZETA)*DLDZT

= X

TO 30

= X

TO 32

40

ZITE (4,60) ZETA,CAPGAM

APTE (6,60) DMP

50

C3S0 = CAPGAM**2

ZZP = XL + C3S0*ZETA

ZPDP = DLDZT + C3S0

ZDSDRT(ZZ**2/(CAPGAM*ZETA)**2)

RETURN

C

FORMAT (3 Chillco MANY ITERATIONS IN WESOLN.10X.SHZETA=1PE15.8.10X.WFS

55
IF (J.L LE.N0 AND LL.LLE.N) Q(1,LL,J)=1
GO TO 105
END CROSS SECTION DATA TO DELETE SPECIES NOT REQUIRED FOR TRANSPORT CALCULATIONS.

110
LO1=1
FOR KO=O OR 12, IT MAY BE NECESSARY TO COMPUTE CROSS SECTIONS FOR SPECIES NOT INCLUDED IN TRANSPORT CALCULATIONS. STATEMENTS 140 TO 300 REVISE THE DATA TO ELIMINATE THEM FROM THE TRANSIT." STATEMENTS 300 THROUGH 390 THEN PROCEED TO EDIT THE CROSS SECTION DATA TO REMOVE SPECIES NOT REQUIRED FOR THE TRANSPORT CALCULATIONS.

CALL 8XSEC (L, MV, LL, J)
IF (L.EQ.0) GO TO 300
IF (J.LE.N0 AND LL.LLE.N) GO TO 120
IF (L.GT.1) LO1=NO(L-1)+1
LO2=NO(L)
IF (LO1.GT.LO2) GO TO 120
DO 130 LO=LO1.LO2
IF (J.LFE.N0 AND IQ(LO).LLE.N) GO TO 140
CONTINUE
GO TO 120
C CROSS SECTIONS ARE REQUIRED FOR THE PAIR (LL,J). FIND UNUSED STORAGE LOCATION FOR THEM.

140
DO 150 JJ=2+N, III=II-1
DO 150 JJ=1, III
IF (Q(1, II, JJ).EQ.O) GO TO 190
CONTINUE

150
NO STORAGE SPACE AVAILABLE. ADD AN ADDITIONAL SPECIES TO CROSS SECTION COMPUTATIONS.

160
IF (N.EQ.20) GO TO S40
N=N+1
JJ=MAX0(J, LL)
DO 160 J=1, J=50
IF (I(J).EQ.N) GO TO 170
CONTINUE

170
DO 180 K=1, 50
IF (I(K).EQ.JJ) GO TO 180
I(J)=J
I(K)=N
GO TO 40
CONTINUE

180
STORE CROSS SECTIONS FOR PAIR (LL,J) IN LOCATION (II, JJ).

190
Q(1,II, JJ)=1.
C IT IS FIRST NECESSARY TO CONVERT ANY KO=10 STEPS FOR THE PAIR TO

200
KQ=12.
NSV=NXO
L=0
LQ1=0
L=L+1
IF (KQ(L).N.E.10) GO TO 270
LQ2=NO(L)
IF (L.GT.1) LO1=NO(L-1)
IF (LQGE.LQ2) GO TO 270
LQ=LQ1
LQ=LQ1
IV=10(LQ)
JG=JG(LQ)
IF ((LLNEJ AND (IVNELL OR JVNEJ)) OR (IVNELL AND JVNEJ))
GO TO 260
CONVERSION TO KQ=12 MUST BE MADE FOR CURRENT STEP.
C FIND LOCATION MM AT WHICH NEW DATA ARE TO BE INSERTED IN V-ARRAY.
NKQ=L
LV=0
CALL BXSECT (LV,MM,K,L)
IF (LVNE0) GO TO 220
REVISE CROSS SECTION DATA
MV=MMV
DO 230 K=MM,MMV
V(MV+4)=V(MV)
MV=MV+1
V(MV+4)=IV
V(MV+4)=JV
MMV=MMV+4
LV=NSV
DO 240 K=NKQ,NSV
KG(LV+1)=KG(LV)
NQ(LV+1)=NQ(LV)
LV=LV+1
NSV=NSV+1
NQ(LV+1)=LQ1
L=LV+2
LV=LQ
DO 250 K=LQ1,LQ
LV=LV-1
IQ(LV+1)=IQ(LV)
JG(LV+1)=JG(LV)
JG(LQ1)=JV
KG(NKQ)=12
CONTINUE
IF (LQLT.LQ2) GO TO 210
CONTINUE
IF (LTNSV) GO TO 200
NKQ=NSV
CONVERSION OF KQ=10 TO KQ=12 COMPLETED. NOW REVISE DATA TO CHANGE.
PAIR (LL,J) TO (11,JJ).
LQ2=NQ(NKQ)
DO 290 Q=1,LQ2
IF (IQ(LQ)NELL OR JQ(LQ)NEJ) GO TO 280
IQ(LQ)=II
JQ(LQ)=JJ
CONTINUE
L=0
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.EQ.JQ(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=IQ(L)
IF (LQ2.LT.LQ1) GO TO 330
DO 320 LQ=LQ1+LQ2
IF (IQ(LQ).NE.0) NEWLO=NEWLQ+1
320 CONTINUE
330 LQ1=LQ2+1
IF (L.EQ.10) GO TO 350
IF (NEWLO.GT.NQ(NKQ)) GO TO 360
MV2=MV1+NV(K)-1
DO 340 MV=MV1+MV2
NEWMV=NEWMV+1
340 V(NEWMV)=MV
350 NEWL=NEWL+1
KO(NEWL)=K
NOQ(NEWL)=NEWLO
GO TO 370
360 NKQ=NEWL+1
370 MV1=NEWL
LL=0
DO 380 LQ=1,400
IF (IQ(LQ).EQ.0) GO TO 380
LL=LL+1
IQ(LL)=IQ(LQ)
JQ(LL)=IQ(LQ)
380 CONTINUE
C COMPUTE UNSPECIFIED NEUTRAL-NEUTRAL CROSS SECTIONS FROM EMPIRICAL
C MIXING LAW. COMPUTATIONS ARE PERFORMED AT LAST SPECIFIED STEP
C WITH KO=10, OR, IF NO STEPS WITH KO=10 ARE SPECIFIED, AFTER ALL
C SPECIFIED CROSS SECTIONS HAVE BEEN COMPUTED.
LL=NKQ+1
DO 390 L=1.NKO
IF (KO(L).EQ.10) LL=L
390 CONTINUE
IF (LL.EQ.NKQ) GO TO 400
KO(LL)=10
NOQ(LL)=NOQ(LL-1)
NKQ=NKQ+1
NN=0
DO 420 L=1,ISS
IF (IELEC.EQ.0) GO TO 410
IZ(L)=L*IJ(J-L+1)
IF (IZ(L).NE.0) GO TO 420
410 NN=NN+1
IF (NN.EQ.L+I.L) GO TO 540
420 CONTINUE
CALL AXSECT (LL+NN)
C FOR IONIZED SPECIES USE EFFECTIVE COULOMB CROSS SECTIONS CHOSEN
C TO FIT TRANSPORT CALCULATIONS OF SPITZER,
IF (LL.EQ.NKQ.AND.NG(II).EQ.NG(LL-1)) GO TO 430
NKQ=NKQ+1
NG(NKQ)=NG(NKQ-1)
430 DO 480 J=1,2
IE=IELEC
KK=J
440 DO 470 JJ=1,2
450 CONTINUE
CALL AXSECT (NKQ,NN)
IF (NG(NKQ).EQ.NG(NKQ-1)) GO TO 460
460 IF (KK.LT.LE+1) GO TO 470
470 CONTINUE
CONTINUE
480 CONTINUE
NN=0
DO 490 L=1,ISS
IF (IZ(L).EQ.9) GO TO 490
490 CONTINUE
CALL AXSECT (NKQ,NN)
IF (NG(NKQ).EQ.NG(NKQ-1)) GO TO 540
C SET UNSPECIFIED ION-NEUTRAL CROSS SECTIONS EQUAL TO TYPICAL VALUE
C ESTIMATED FOR N-O+ INTERACTION.
DO 500 L=1,ISS
500 CONTINUE
CALL AXSECT (NKQ,II+1)
IF (NG(NKQ-1).EQ.NG(NKQ-1)) GO TO 510
NKQ=NKQ-1
GO TO 520
* 
V(NEWMV+1)=996.
V(NEWMV+2)=5.
* 
CONTINUE
IF (NKQ*GT*100) GO TO 540
\* 
RETURN
C INSUFFICIENT CROSS SECTION DATA AVAILABLE. TERMINATE CASE.
\* 
540 \* 
WRITE (6,550) 
ERR=TRUE
WRITE (6,XSDMP)
GO TO 530
\* 
\* 
550 \* 
FORMAT (/116HOTRANSPORT PROPERTIES OF DESIRED MIXTURE CANNOT BE CALCULATED FROM AVAILABLE DATA. REVISE CROSS SECTION INPUT DATA. )
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 DØ 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, ICOND(J).

a. ICOND(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(J) = 2), and must be specified as a throat
section (ICOND(J) = 1). For such a throat section, dy/dx
must be zero at x = 0. Thus, equation (122) shows that
the abscissa P2 of the circle center must be zero.

\[ P_2 = 0 \]

The circle radius is equal to PAR(1,J):

\[ P_3 = 2.54 \cdot PAR(1,J) \]  (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):

\[ P_1 = P_3 + \frac{1}{2} \cdot 2.54 \cdot DTH \]  (180)

where DTH is the throat diameter in inches.

b. ICOND(J) = 2

For the condition ICOND(J) = 2, the current profile
section is assumed to be straight (ISHAPE(J) = 1) and tangent
to an adjacent circular section nearer the throat. The in-
dex 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 \( R = \text{PARAM}(3, \text{INDEX}) \) 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:

\[ X_c = \text{PARAM}(2, \text{INDEX}) \quad (183c) \]

\[ Y_c = \text{PARAM}(1, \text{INDEX}) \quad (183d) \]
Figure 29. Geometry of the Condition ICOND(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\(\text{OND}(J) = 3\)

For the condition IC\(\text{OND}(J) = 3\), 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 \text{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
Figure 30. Geometry of the Condition: $ICOND(J) = 3$
The condition ICOND(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 $X_1 = 2.5A \cdot \text{PAR}(2, J)$ 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 $\phi$. This angle is given by

$$\phi = \frac{1}{2} (\pi - |\theta_2 - \theta_1|)$$  \hspace{1cm} (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 $C$,

$$L = R / \sin \phi$$  \hspace{1cm} (191)

For ISHAPE(J) = 2 (parts (a) and (c) of figure 31), the angle CIA is equal to $\phi - \theta_1$. For ISHAPE(J) = 3 (parts (b) and (d) of figure 31), angle CIA is equal to $\phi - \theta_2$. Hence, for ISHAPE(J) = 2,

$$X_C = X_1 - S_1 L \cos (\phi - \theta_1)$$  \hspace{1cm} (192a)

$$Y_C = Y_1 + L \sin (\phi - \theta_1)$$  \hspace{1cm} (192b)

For ISHAPE(J) = 3,

$$X_C = X_1 + S_1 L \cos (\phi - \theta_2)$$  \hspace{1cm} (193a)
Upstream, ISHAPE(J) = 2

Downstream, ISHAPE(J) = 2

Upstream, ISHAPE(J) = 3

Downstream, ISHAPE(J) = 3

Figure 31. Geometry of the Condition: LODND(J) = 4
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 $L \cos \phi$.

The projection of this distance onto the X-axis is $L \cos \phi \cdot \cos \theta_1$. Hence, the abscissa of the junction point is

$$X_p = X_1 - S_1 L \cos \phi \cos \theta_1$$  \hspace{1cm} (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 NØZFIT 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*, NØZFIT 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 IWØRDL, 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, NØZFIT

*See discussion of NØZFIT 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 (1.21), (1.22), or (1.23) 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 $PLOTS$ 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 $N_	ext{ZFIT}$. Two versions of this routine are provided (Section A.3), one for each type of computer system.

The IBM 360 version sets the indicator $IWORDL$ to 2, to induce $N_	ext{ZFIT}$ 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 $PLOTS$ 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 $IWORDL$ to 1, to induce $N_	ext{ZFIT}$ 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 DXYV (references 3, 17) is called for each axis to determine the arguments DX, DY, NXX, NYY, III, JJJ, NXY of subroutine GRIDLV. If either axis is to be used as a logarithmic scale (LX \( \neq 0 \) or LY \( \neq 0 \)), then the corresponding call to DXYV 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 DXYV 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 GRIDLV 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 NXV and NYV are used to initialize IX1 and 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 NSYMBl is set equal to ICURVE, so that a new symbol from the standard table in POINTB is used for the points on each successive curve in a family of curves. If NC(1) is positive, NSYMBl 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(1) 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 LABLV (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 (XL, 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 COSA 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 DO loop, DO 70 I = 1, NP, the quantities VX1 and W1 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>V</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>@</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 DO 120 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>y</td>
</tr>
<tr>
<td>8</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>i</td>
</tr>
<tr>
<td>10</td>
<td></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

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSL(I)</td>
<td>Array containing alphanumeric abscissa label for plots</td>
</tr>
<tr>
<td>ACØN</td>
<td>Conversion factor ((180/\pi)) from radians to degrees</td>
</tr>
<tr>
<td>ATP(JP)</td>
<td>Axial coordinate at the downstream boundary of the JPth profile section (cm)</td>
</tr>
<tr>
<td>CARDS</td>
<td>Logical control; .FALSE. value suppresses punched card output</td>
</tr>
<tr>
<td>COMMENT(I)</td>
<td>Hollerith descriptive data</td>
</tr>
<tr>
<td>CTHETA</td>
<td>(\cos(\text{THETA}))</td>
</tr>
<tr>
<td>DTH</td>
<td>Throat diameter (inch)</td>
</tr>
<tr>
<td>DX</td>
<td>Increment in X for calculation of profile (inch)</td>
</tr>
<tr>
<td>ENDJØB</td>
<td>Logical control; .TRUE. value terminates the job at the end of the current case</td>
</tr>
<tr>
<td>FISHAP(I)</td>
<td>Floating point value representing ISHAPE(I) in punched-card output</td>
</tr>
<tr>
<td>FNSECT(I)</td>
<td>NSECTS(I) + 0.1 for card output</td>
</tr>
<tr>
<td>I</td>
<td>DØ index</td>
</tr>
<tr>
<td>IC</td>
<td>ICØND(J) for the Jth profile section</td>
</tr>
<tr>
<td>ICØND(J)</td>
<td>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)</td>
</tr>
<tr>
<td>ICUROVE</td>
<td>Unused argument of subroutine GRAPH</td>
</tr>
<tr>
<td>INCR(L)</td>
<td>0 for (L = 1); 1 for (L = 2)</td>
</tr>
</tbody>
</table>
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)

IWØRDL

Indicator for number of characters per word on the type of computer system being used:

- IWØRDL = 1 6-character word (UNIVAC 1108)
- IWØRDL = 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
LINE(I)
LL
NC(I)
NOZZLE
NS
NSECTS(L)
NSM1
NSECTS(L)
ØRDL(I)
PAR(I,J)
PARAM(I,J)
PHI
PI
PLABL(I)
PLOTS
PVAL

Indicator for upstream profile sections (L = 1) and downstream sections (L = 2)
Argument of subroutine GRAPH (see below)
Argument of subroutine GRAPH (see below)
Argument of subroutine GRAPH (see below)
Profile index for use in NATA
Number of profile sections upstream (L = 1) and downstream (L = 2) of the throat
Alphanumeric label for ordinate in plot
Input parameter values for the Jth profile section; see discussion of inputs, Appendix D of Volume II
Parameters of curvefit for Jth profile section; see discussion of outputs, Appendix D of Volume II
0.5 (\(\pi - |\Theta_{3} - \Theta_{1}|\)); see discussion of ICOND = 4 in Section A.1.1.
3.14159265
Argument of subroutine GRAPH (see below)
Logical control, .FALSE. value suppresses plot output
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 ICØND = 4, inclination angle of the adjacent section closer to the throat

THETA2  For ICØND = 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 ICØND = 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

**X1** 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)

**IWØRDL** Defined above (Main program)

**PLØTS** 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

ABSL
Array containing 48-character label for abscissa scale

CURVE
ICURVE expressed as a floating-point value

DC
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

DX
Floating point data increment at which vertical grid lines are displayed

DY
Floating point data increment at which horizontal grid lines are displayed

DYL
Amount by which YL is decreased and YU increased to allow space for numerical labels of curves

HYP
Distance (in raster units) between the first two points of a curve

I
IØ index

ICURVE
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.

IDXV
Interval in raster units between the abscissas of the first two points in a curve

IDYV
Interval in raster units between the ordinates of the first two points of a curve
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IERR</td>
<td>Error indicator in subroutine DXYDV (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</td>
</tr>
<tr>
<td>III</td>
<td>Argument of subroutine GRIDIV (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</td>
</tr>
<tr>
<td>IMAG(I)</td>
<td>For I = 1 and 2, log₁₀ of the largest numerical value to be plotted on the abscissa and ordinate, respectively</td>
</tr>
<tr>
<td>IPVM</td>
<td>log₁₀ of the absolute magnitude of a parameter value (PVAL)</td>
</tr>
<tr>
<td>IXI</td>
<td>Abscissa of the location where a curve label is to be printed (raster units)</td>
</tr>
<tr>
<td>IY</td>
<td>Ordinate of the location where a parameter value is to be printed in the margin (raster units)</td>
</tr>
<tr>
<td>IY1</td>
<td>Ordinate of the location where a curve label is to be printed (raster units)</td>
</tr>
<tr>
<td>IY2</td>
<td>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)</td>
</tr>
<tr>
<td>I2</td>
<td>( I + 1 )</td>
</tr>
<tr>
<td>J</td>
<td>DP index</td>
</tr>
<tr>
<td>JCURVE</td>
<td>Counter for the total number of curves plotted in a given frame</td>
</tr>
</tbody>
</table>
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 SMXYV (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 PØINTB)

NC(1) = 0 Suppress plotting of points

NC(1) > 0 Use the NC(1)th character in the standard table (see PØINTB)

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)

NMAX

Argument of subroutine LABLV (see references 3, 17); maximum number of characters to be displayed to the left of the decimal point
NF  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

NLAST  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

NP  Number of data points to be plotted

NPL  Counter for families of curves plotted in the same frame

NSYMNL  Index of point symbol in standard table (see POINTB)

NXVL  Abscissa of first point in raster coordinates

NXX  Argument of subroutine DXDyV (references 3, 17); not used in GRAPH

NXY(I)  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)

NYVL  Ordinate of first point in raster coordinates

NYY  Argument of subroutine DXDyV (references 3, 17); not used in GRAPH

ØRDL(I)  Alphanumeric label for ordinate

PLABL(I,J)  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

COSA | 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 PDINTV (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
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RASTER</td>
<td>Logical indicator, .TRUE. for entry through LINEBR, .FALSE. for entry through LINEB</td>
</tr>
<tr>
<td>SINA</td>
<td>Sine of the angle between the vector to be drawn and the positive X-axis</td>
</tr>
<tr>
<td>SQ</td>
<td>Square of the distance between the start and end points of the vector (raster units)</td>
</tr>
<tr>
<td>VX1</td>
<td>Floating point value of NX1</td>
</tr>
<tr>
<td>VY1</td>
<td>Floating point value of NY1</td>
</tr>
<tr>
<td>X1</td>
<td>Abscissa of the start point</td>
</tr>
<tr>
<td>X2</td>
<td>Abscissa of the end point</td>
</tr>
<tr>
<td>Y1</td>
<td>Ordinate of the start point</td>
</tr>
<tr>
<td>Y2</td>
<td>Ordinate of the end point</td>
</tr>
</tbody>
</table>

A.2.5 Subroutine PØINTB

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR(I)</td>
<td>Array of characters for plotting points</td>
</tr>
<tr>
<td>IDARK</td>
<td>Argument of RITE2V controlling darkness of plotted symbol (see references 3, 17)</td>
</tr>
<tr>
<td>IX</td>
<td>Abscissa of location where point is plotted (raster coordinate)</td>
</tr>
<tr>
<td>IY</td>
<td>Ordinate of location where point is plotted (raster coordinate)</td>
</tr>
<tr>
<td>K(I)</td>
<td>Argument of RITE2V controlling orientation of plotted symbol; used to present each of the symbols &quot;H&quot;, &quot;C&quot;, and &quot;Y&quot; in two orientations</td>
</tr>
<tr>
<td>NLAST</td>
<td>Argument of RITE2V (references 3, 17); not used in PØINTB</td>
</tr>
</tbody>
</table>
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 (Avcc) and UNIVAC (NASA/JSC) versions of FRAMEB.
**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 an integer.
- **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
- **PARAM(J)**: Parameter values for J-th section.
  - For ISHAPE(J)=1, PARAM(J)=angle of inclination to nozzle axis in degrees (positive value).
  - For ISHAPE(J)=2 or 3, PARAM(J)=circle radius in inches.
- **DTH**: Throat diameter in inches.
- **ACON=180.*PI**: throttle condition defining J-th section.
  - **ICOND(J)**: Inlet condition defining J-th section (Y=1.27*0 and Y/D=0.
  - **AT X=0**:
    - **ICOND(J)=2**: Straight section (J) is tangent to adjacent circular section near the throat.
    - **ICOND(J)=3**: Circular section (J) is tangent to adjacent straight section (nearer to the throat).
  - **AT X=2.5A*PAR(2)**:
    - **ICOND(J)=4**: Circular section (J) is used to break sharp angle between two adjoining straight sections which intersect at X=2.5A*PAR(2).
- **XSTART**: Left limit on X for calculating nozzle profile (inch).
- **ZEROC**: Inlet position in inches above the throat (negative).
- **NOZZLE INDEX FOR NATA**: Logical control - false value suppresses punch output.
- **PLOTS**: Logical control - false value suppresses plots.
- **ENDJOB**: Logical control for job termination.

**Variables and Constants**

### LOGICAL
- **ENDJOB, CARDS, PLOTS**
- **PARAM(3:12)**, **ATP(11)**, **PLABL(6)**
- **DIMENSION ISHAPE(12)**, **DIMENSION FNSEC(2)**, **FN2AP(12)**
- **DATA PI/3.14159265/, CARDS, PLOTS /2*TRUE*/
- **DIMENSION PAR(2:12)**, **ICOND(12)**, **NSEC(2)**
- **DIMENSION INCRA(2)**, **S2ARR(3)**

**Name List**

- **INPUT**: NSEC, ISHAPE, PAR, ICOND, DTH, ENDO, XSTART, 1, ZEROC, NOZZLE, CARDS, PLOTS
GO TO 30
READ (3,340) (CMMNT(I),I=1,12)
C FIRST 4 CHARACTERS ON THIS CARD SHOULD BE NAME OF FACILITY FOR USE
C IDENTIFICATION IN NATO (DCA, EOS, ETC)
READ (5,INPUT)
*RITE (6,INPUT)
C ZERO ARRAYS
DO 40 I=1,12
IF (I*NE.12) ATRP(I)=0.
DO 40 J=1,3
PAR(X,J)=0.
40 CONTINUE
C COMPUTE PROFILE FIT PAPAMETERS
DO 140 L=1,2
L=1 UPSTREAM
L=2 DOWNSTREAM
NS=NSECTS(L)
DO 130 K=1,NS
IF (L.EQ.2) GO TO 50
31=1.
J=NS-K+1
ISGN=1
GO TO 60
50 J=NSECTS(1)+K
ISGN=-1
S1=1.
60 IC=ICOND(J)
JP=J-INCAL(L)
INDEX=J+ISGN
GO TO (70,20,90,100)* IC
70 IF (ISHAPE(J)*NE.2) GO TO 120
C THRACAT SECTION (ICOND=1)
PARAM(3,J)=2.*S4*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=S2AR(ISH)
XC=PARAM(2,INDEX)
YC=PARAM(1,INDEX)
STHETA=SIN(THETA)
CTHETA=COS(THETA)
TTHETA=STHETA/CTHETA
X1=XC-S1*S2*R*STHETA
Y1=S2*Y*CTHETA+YC
PARAM(I,J)=Y1-S1*X1*TTHETA
PARAM(2,J)=S1*TTHETA
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 ICOND=3
R=2*S4*PAR(1,J)
X1=2*S4*PAR(2,J)
J=-ISHAPE(J)
S2=S2*PAR(ISH)
THETA=PAR(1,INDEX)/ACON
Y1=PAR(1,INDEX)*PAR(2,INDEX)*X1
PARAM(1,J)=Y1-S2*R*COS(THETA)
PARAM(2,J)=X1+S1*S2*R*SIN(THETA)
PARAM(3,J)=R
ATP(JP)=X1
GO TO 130
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 ICOND=4
R=2*S4*PAR(1,J)
X1=2*S4*PAR(2,J)
THETA=PAR(1,INDEX)/ACON
THETA=PAR(1,INDEX2)/ACON
PHI=3*S4*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
X1=X1/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(PHI-THETA1)
PARAM(1,J)=-Y1*X1*SIN(PHI-THETA1)
GO TO 133
110 PARAM(2,J)=X1+S1*X1*COS(PHI-THETA2)
PARAM(1,J)=-Y1-X1*SIN(PHI-THETA2)
GO TO 130
120 WRITE (6,350) J
CALL EXIT
130 CONTINUE
140 CONTINUE
C
OUTPUT
C
PRINTED OUTPUT
IF (IORDL*EQ.1) GO TO 150
WRITE (6,350) (COMMNT(I),I=1,18)
GO TO 160
150 WRITE (6,370) (COMMNT(I),I=1,12)
160 WRITE (6,380)
WRITE (6,390)
NS=NSECTS(1)+NSECTS(2)
GO 180 J=1:NS
NOZ 111
NOZ 112
NOZ 113
NOZ 114
NOZ 115
NOZ 116
NOZ 117
NOZ 118
NOZ 119
NOZ 120
NOZ 121
NOZ 122
NOZ 123
NOZ 124
NOZ 125
NOZ 126
NOZ 127
NOZ 128
NOZ 129
NOZ 130
NOZ 131
NOZ 132
NOZ 133
NOZ 134
NOZ 135
NOZ 136
NOZ 137
NOZ 138
NOZ 139
NOZ 140
NOZ 141
NOZ 142
NOZ 143
NOZ 144
NOZ 145
NOZ 146
NOZ 147
NOZ 148
NOZ 149
NOZ 150
NOZ 151
NOZ 152
NOZ 153
NOZ 154
NOZ 155
NOZ 156
NOZ 157
NOZ 158
NOZ 159
NOZ 160
NOZ 161
NOZ 162
NOZ 163
NOZ 164
NOZ 165
GO TO 300
Y=PARAM (1.K)+SORT(PAPAM(3.K)**2-(X-PAR(2,K))**2)
310 CONTINUE
IF (*.NOT.PLOTS) GO TO 320
320 IF (*.NOT.ENDJOB) GO TO 10
IF (PLOTS) CALL FRAM92
CALL XIT
SUBROUTINE FRAMFB (IMCOREP, PLOTS)
LOGICAL PLOTS
FRAME FOR PLOTS AND CLOSES PLOT DATA
SET IMCOREP, XNAVE(3), XMEMO, XMEMO
DATA XNAVE, XMEMO, XNAME, XMEN
IF (PLOTS) CALL 10FRMV (XNAME, XMEMO)
RETURN
END
SUBROUTINE GRAPH (NF, LX, LY, NP, X, Y, XL, YL, DU, DC, NC, LINE, CRDL, ABSL, GRA
1, TITL, LL, PV AL, PLABL, I CURVE)
DIMENSION X(120), Y(120), LINE(112), CRDL(112), ABSL(112), TITL(112),
1 PLABL(132), XNY(2), IMAG(2), XNXY(2), NC(2)
C
NOTE - 48 CHARACTERS IN AUSCISSA, ORDINATE, AND TITLE LABELS
C
NOTE - CALL SETMIV (30,125,24,24) BEFORE FIRST CALL TO GRAPH,
C
IF PARAMETER TABLE IS TO BE PRINTED

10 IF (NP .EQ. 0) GO TO 110
  JCURVE=0
  NPL=0
  CALL SMXYV (LX, LY)
  IF (LX .EQ. 0) GO TO 10
  DX=1
  NNX=1
  GO TO 20

20 CALL DXYV (1, XL, XU, DX, NNN, ITI, NXX, DC, I ERR)
  DY=1
  MMM=1
  GO TO 40

30 CALL DXYV (2, YL, YU, DY, MMM, JJJ, NYY, DC, I ERR)
  IF (YX .EQ. 0) GO TO 50

40 IF (IF .EQ. 0) GO TO 50

50 IF (IMAG(1) .EQ. I - 1, 2) GO TO 80

60 NXY(I) = IABS(I M A)(I) + 4
  GO TO 90

70 NXY(I) = I M A(I) + 1
  GO TO 90

80 NXY(I) = - 3
  CONTINUE

90 NXY(I) = - 3

100 CALL GRIDIXV (1, XL, XU, YL, YU, DX, DY, NNN, MMM, III, JJJ, NXY(1), NXY(2))
  I CURVE=0

110 I CURVE=ICURVE+1
  IF (ICURV .EQ. 0) NPL=NPL+1
  JCURVE=JCURVE+1
  I X=NXY(X(1))
  I Y=NXY(Y(1))
  NC=NC(2)
  NC=NC(1)
  IF (NC) 120, 140, 130
120 NSYMBL=ICURVE
   GO TO 140
   GRA 56
130 NSYMBL=NC1
140 GO 190 I=1, NP
150 IF (NC1.EQ.0) GO TO 150
   GRA 59
   GRA 60
   CALL PCINTB (IX1,IV1,NSYMBL,NC2)
   GRA 61
150 IF (I.EQ.NP) GO TO 180
   GRA 62
   I2=I+1
160 IF (LINE(I)) 170,170,160
   GRA 63
   CALL LINEB (X(I),Y(I),X(I2),Y(I2),LINE)
170 IX1=NXVIX(I2)
   GRA 64
   IY1=NYVY(I2)
   GRA 65
   CONTINUE
   GRA 66
180 IF (JCURVE.GT.1) GO TO 190
   CALL CHSIZV (3,3)
   CALL RITEV (80,9,1023,90,2,48,1,ABSL,NLAST)
   CALL RITE2V (9,60,1023,180,2,48,1,ORDL,NLAST)
   CALL PITE2V (80,1012,1023,90,2,48,1,TITL,NLAST)
   GRA 67
190 IF (LL.EQ.0) RETURN
   CURVE=ICURVE
   GRA 68
   IF (LL.EQ.1) GO TO 200
   GRA 69
   NXV1=NXV(X(1))
   GRA 70
   NYV1=NYV(Y(1))
   GRA 71
   IDXV=NXV(X(I2))=NXV1
   GRA 72
   IDYV=NYV(Y(I2))=NYV1
   GRA 73
   HYP=IDXV*2+IDYV**2
   GRA 74
   HYP=SQRT(HYP)
   GRA 75
   IX1=NXV1-16.*IDXV/HYP
   GRA 76
   IY1=NYV1-16.*IDYV/HYP
   GRA 77
   CALL LADLV (CURVE,IX1,IY1,2,2,2)
   GRA 78
200 IF (ICURVE.GT.1) GO TO 230
   IY=950.-NPL-1.*300
   GRA 79
   IY=IY+15
   GRA 80
   CALL LINEBR (930.,IY2,1023.,IY2,LINE)
   GRA 81
   GO 220 I=1,2
   GRA 82
   GO 210 J=1,2
210 CALL PRINTV (12,PLABL(I,I),924.,IY)
   GRA 83
220 IY=IY-20
   GRA 84
230 IY=IY-20
   GRA 85
   IF (LL.EQ.1) GO TO 240
   CALL LADLV (CURVE,950.,IY,2,1,2)
   GRA 86
240 CALL PCINTB (924.,IY,NSYMBL,NC2)
   GRA 87
   IF (PVAL.EQ.0) GO TO 250
   GRA 88
   IPVM=ALUG10(ABS(PVAL))
   GRA 89
   GO TO 260
   GRA 90
250 IPVM=0
260 IF (IPVM.GT.4.OR.IPVM.LT.-2) GO TO 290
   GRA 91
270 IF (IPVM) 270,270,280
   GRA 92
270 NCHAR=IABS(IPVM)+5
   GRA 93
   NMAX=1
   GRA 94
   GO TO 300
   GRA 95
   RCHAR=5
   GRA 96
   RMAX=IPVM+1
   GRA 97
290 NMAX=1
   GRA 98
   GO TO 300
   GRA 99
290 NMAX=1
   GRA 100
   GO TO 300
   GRA 101
   NCHAR=5
   GRA 102
   NMAX=IPVM+1
   GRA 103
290 NMAX=1
   GRA 104
   GO TO 300
   GRA 105
   NCHAR=5
   GRA 106
   NMAX=IPVM+1
   GRA 107
290 NMAX=1
   GRA 108
   GO TO 300
   GRA 109
   NMAX=1
   GRA 110
NCHAR=3
CALL LABELV (PVAL,940,IY,NCHAR,1,NDMAX)
RETURN

FORMAT (S1H0GRID GENERATION IMPOSSIBLE WITH THE PARAMETERS
XL=,IZEPAGRA 116
110.3,SH, XU=,E10.3,5H, YL=,E10.3,5H, YU=,E10.3)
END
SUBROUTINELINEB(X1,Y1,X2,Y2,LLINE)  LIN  1
LILE=LLINE(1), IDARK=LLINE(2), IPAR=LLINE(3)
LIN  2
X1  = ABSCISSA OF FIRST POINT  LIN  3
LIN  4
Y1  = ORDIATE OF FIRST POINT  LIN  5
LIN  6
X2  = ABSCISSA OF SECOND POINT  LIN  7
LIN  8
Y2  = ORDIATE OF SECOND POINT  LIN  9
LINE =  1 CONNECT POINTS WITH A CONTINUOUS LINE
LIN  10
=  2 CONNECT POINTS WITH A DASHED LINE
LIN  11
=  3 CONNECT POINTS WITH A STRING OF CHARACTERS
LIN  12
IF LINE=1 IPAR IS IGNORED
LIN  13
IF LINE=2 IPAR = LENGTH OF DASHES IN MILLIMETERS
LIN  14
IF LINE=3 IPAR = INTEGER (NS) SELECTING PLOT SYMBOL FROM
LIN  15
POINT TABLE
LIN  16
IDARK = NUMBER OF TIMES LINE OR SYMBOLS ARE DRAWN
LIN  17
LOGICAL RASTER
LIN  18
DIMENSION LLINF(3)
LIN  19
RASTER=FALSE
LIN  20
GO TO 1C
LIN  21
ENTRY LINEBR(MX1,MY1,MX2,MY2,LLINE)
LIN  22
RASTER=TRUE
LIN  23
LINE=LLINE(1)
LIN  24
IF (LINE.LE.0) RETURN
LIN  25
IDARK=LLINE(2)
LIN  26
IPAR=LLINE(3)
LIN  27
IF (.NOT.RASTER) GO TO 20
LIN  28
NX1=MX1
LIN  29
NX2=MX2
LIN  30
NY1=MY1
LIN  31
NY2=MY2
LIN  32
GO TO 3C
LIN  33
20  *X1=NX1*X1
LIN  34
NX2=NX2*X2
LIN  35
NY1=NY1*Y1
LIN  36
NY2=NY2*Y2
LIN  37
GO TO 50
LIN  38
30  IF (LINE.GT.1) GO TO 50
LIN  39
DO 40 I=1,IDARK
LIN  40
CALL LINEV(NX1,NY1,NX2,NY2)
LIN  41
RETURN
LIN  42
50  SQ=(NX2-NX1)**2+(NY2-NY1)**2
LIN  43
DIST=SQR(T(SQ)
LIN  44
VX1=NX1
LIN  45
VY1=NY1
LIN  46
COSA=(NX2-NX1)/DIST
LIN  47
SINA=(NY2-NY1)/DIST
LIN  48
IF (LINE.GT.2) GO TO 8C
LIN  49
DEL=10.*IPAR
LIN  50
NP=DIST/DL
LIN  51
HOEL=0.*DEL
LIN  52
IDX=HOEL*COSA
LIN  53
IDY=HOEL*SINA
LIN  54
DO 70 I=1,NP
LIN  55
DO 60 J=1,IDARK
LIN  56
70  VX1=VX1+DEL*COSA
LIN  57
CALL LINE2V(NX1,NY1,IDX,IDY)
LIN  58
IF (I.EQ.NP) GO TO 70
LIN  59
VX1=VX1+DEL*COSA
LIN  60
CALL LINE2V(NX1,NY1,IDX,IDY)
LIN  61
IF (I.EQ.NP) GO TO 70
LIN  62
VY1=VY1+DEL*SINA
NX1=VX1
NY1=VY1
70 CONTINUE
RETURN
80 IF (IPAR*EQ.0) GO TO 90
DEL=15,
GO TO 100
90 DEL=5,
100 NP=DIST/DEL
DO 120 I=1,NP
DO 110 J=1,IPARK
110 CALL POINTV (NX1,NY1,-IPAR,1*)
IF (I.EQ.NP) GO TO 120
VXI=VXI+DEL*COSA
VY1=VY1+DEL*SINA
NX1=VX1
NY1=VY1
120 CONTINUE
RETURN
END
REFERENCES


