Technical Report 32-1240

Addendum

ELAS—A General-Purpose Computer Program for the Equilibrium Problems of Linear Structures

Volume II. Documentation of the Program

Senol Utku

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 15, 1969
NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.
Preface

The work described in this report was performed by the Engineering Mechanics Division of the Jet Propulsion Laboratory.

The program was developed by Dr. Senol Utku and Dr. Fevzican A. Akyuz, and is dedicated to the memory of Professor M. Inan of the Technical University of Istanbul.
Acknowledgment

The author is indebted to Vivia Crew for her help in editing all documents related with the ELAS program.
Contents

I. Introduction .......................... 1

II. Main Program and Subroutines of Link 1 ................................ 2
   A. Main Program of Link 1 ................................ 2
   B. Subroutines of Link 1 ................................ 3
      1. Subroutine ARAN ................................. 3
      2. Subroutine BUNG ................................. 3
      3. Subroutine COOR ................................. 3
      4. Subroutine CORG ................................. 3
      5. Subroutine EXCH ................................. 3
      6. Function LEBIN ................................. 4
      7. Subroutine SEBIN ................................. 4
      8. Subroutine MESG ................................. 4
      9. Subroutine MEST ................................. 4
     10. Subroutine OUTPT .............................. 4
     11. Subroutine SRAT ............................... 4
     12. Subroutine TABL ............................... 5
     13. Subroutine TICK ............................... 5
     14. Subroutine TOPO ............................... 5

III. Main Program and Subroutines of Link 2 ................................. 6
   A. Main Program of Link 2 ............................. 6
   B. Subroutines of Link 2 .............................. 7
      1. Subroutine ADM ................................. 7
      2. Subroutine BEAM ................................. 8
      3. Subroutine CAS2 ................................. 8
      4. Subroutine CODI ................................. 8
      5. Subroutine CORT ................................. 8
      6. Subroutine CUTE ................................. 8
      7. Subroutine DARN ................................. 8
      8. Subroutine DMM ................................. 9
Contents (contd)

9. Subroutine ELDI ........................................ 9
10. Subroutine PLBE ........................................ 9
11. Subroutine RLOC ....................................... 9
12. Subroutine S01 ........................................ 9
13. Subroutine S02 ........................................ 9
14. Subroutine S03 ........................................ 10
15. Subroutine S04 ........................................ 10
16. Subroutine S05 ........................................ 10
17. Subroutine S07 ....................................... 11
18. Subroutine S09 ........................................ 11
19. Subroutine S11 ....................................... 11
20. Subroutine S13 ....................................... 11
21. Subroutine S15 ....................................... 12
22. Subroutine S17 ....................................... 12
23. Subroutine S18 ....................................... 12
24. Subroutine STFS .................................... 12
25. Subroutine STRA .................................... 12
26. Subroutine Tick ..................................... 13
27. Subroutine TOPO ..................................... 13
28. Subroutine TRAN ..................................... 13
29. Subroutine TRIM ..................................... 13
30. Subroutine TRM ..................................... 13

IV. Main Program and Subroutines of Link 3 .................. 14
   A. Main Program of Link 3 ............................... 14
   B. Subroutines of Link 3 .............................. 15
      1. Subroutine ELST ................................ 15
      2. Subroutine PUNC ................................ 15
      3. Subroutine RESI ................................ 15
      4. Subroutine RESW ................................ 15
      5. Subroutine TICK ................................ 16
      6. Subroutine VELAS ............................... 16
## Contents (contd)

### V. Main Program and Subroutines of Link 4

#### A. Main Program of Link 4

- Subroutine ABEQ .................................. 17
- Subroutine AGEL .................................. 17
- Subroutine BEST .................................. 17
- Subroutine BOFI .................................. 17
- Subroutine CAS4 .................................. 17
- Subroutine CODI .................................. 17
- Subroutine DIMI .................................. 17
- Subroutine DINA .................................. 17
- Subroutine EPAN .................................. 17
- Subroutine FINDQ .................................. 17
- Subroutine FINDX .................................. 17
- Subroutine GENE .................................. 17
- Subroutine INER .................................. 17
- Subroutine INLZ .................................. 17
- Subroutine INV .................................. 17
- Subroutine LEST .................................. 17
- Subroutine MDIN .................................. 17
- Subroutine META .................................. 17
- Subroutine QUAD .................................. 17
- Subroutine REVO .................................. 17
- Subroutine ROTA .................................. 17
- Subroutine SAME .................................. 17
- Subroutine SCAL .................................. 17
- Subroutine SETA .................................. 17
- Subroutine STRA .................................. 17
- Subroutine STRS .................................. 17
- Subroutine TEMP .................................. 17
- Subroutine TICK .................................. 17
- Subroutine TOPO .................................. 17
- Subroutine TRAN .................................. 17

#### B. Subroutines of Link 4

- Subroutine ABEQ .................................. 17
- Subroutine AGEL .................................. 17
- Subroutine BEST .................................. 17
- Subroutine BOFI .................................. 17
- Subroutine CAS4 .................................. 17
- Subroutine CODI .................................. 17
- Subroutine DIMI .................................. 17
- Subroutine DINA .................................. 17
- Subroutine EPAN .................................. 17
- Subroutine FINDQ .................................. 17
- Subroutine FINDX .................................. 17
- Subroutine GENE .................................. 17
- Subroutine INER .................................. 17
- Subroutine INLZ .................................. 17
- Subroutine INV .................................. 17
- Subroutine LEST .................................. 17
- Subroutine MDIN .................................. 17
- Subroutine META .................................. 17
- Subroutine QUAD .................................. 17
- Subroutine REVO .................................. 17
- Subroutine ROTA .................................. 17
- Subroutine SAME .................................. 17
- Subroutine SCAL .................................. 17
- Subroutine SETA .................................. 17
- Subroutine STRA .................................. 17
- Subroutine STRS .................................. 17
- Subroutine TEMP .................................. 17
- Subroutine TICK .................................. 17
- Subroutine TOPO .................................. 17
- Subroutine TRAN .................................. 17
Contents (contd)

31. Subroutine UNIT .................................................. 24
32. Subroutine VECT .................................................... 24

VI. Semidetailed Flowcharts ........................................... 25

VII. Source Program Listings .......................................... 112

References ....................................................................... 161

Tables

 V-1. Values of important parameters used in subroutine ABEQ for various classes ........................................... 19
 V-2. Arrangement of prescribed boundary forces by subroutine ABEQ in SR vector for the eight class types .............. 19
 VII-1. Source program listing of main program of Link 1 (input link) .......................................................... 113
 VII-2. Source program listing of subroutine ARAN (Link 1) ................................................................. 115
 VII-3. Source program listing of subroutine BUNG (Link 1) ........................................................................ 116
 VII-4. Source program listing of subroutine COOR (Link 1) ........................................................................ 117
 VII-5. Source program listing of subroutine CORG (Link 1) ........................................................................ 117
 VII-6. Source program listing of subroutine EXCH (Link 1) ........................................................................ 117
 VII-7. Source program listing of function LEBIN and subroutine SEBIN (Link 1) ......................................... 117
 VII-8. Source program listing of subroutine MESG (Link 1) ........................................................................ 118
 VII-9. Source program listing of subroutine MEST (Link 1) ........................................................................ 118
 VII-10. Source program listing of subroutine OUTPT (Link 1) ..................................................................... 118
 VII-11. Source program listing of subroutine SRAT (Link 1) ....................................................................... 119
 VII-12. Source program listing of subroutine TABL (Link 1) ....................................................................... 120
 VII-13. Source program listing of subroutine TICK (Link 1) ....................................................................... 120
 VII-14. Source program listing of subroutine TOPO (Link 1) ..................................................................... 120
 VII-15. Source program listing of main program of Link 2 (generation link) ................................................... 121
 VII-16. Source program listing of subroutine ADM (Link 2) ....................................................................... 123
 VII-17. Source program listing of subroutine BEAM (Link 2) ..................................................................... 123
 VII-18. Source program listing of subroutine CAS2 (Link 2) ...................................................................... 123
 VII-19. Source program listing of subroutine CODI (Link 2) .................................................................... 123
 VII-20. Source program listing of subroutine CORT (Link 2) .................................................................... 124
### Contents (contd)

#### Tables (contd)

<table>
<thead>
<tr>
<th>VII-21.</th>
<th>Source program listing of subroutine CUTE (Link 2)</th>
<th>124</th>
</tr>
</thead>
<tbody>
<tr>
<td>VII-22.</td>
<td>Source program listing of subroutine DARN (Link 2)</td>
<td>125</td>
</tr>
<tr>
<td>VII-23.</td>
<td>Source program listing of subroutine DMM (Link 2)</td>
<td>125</td>
</tr>
<tr>
<td>VII-24.</td>
<td>Source program listing of subroutine ELDI (Link 2)</td>
<td>125</td>
</tr>
<tr>
<td>VII-25.</td>
<td>Source program listing of subroutine PLBE (Link 2)</td>
<td>126</td>
</tr>
<tr>
<td>VII-26.</td>
<td>Source program listing of subroutine RLOC (Link 2)</td>
<td>126</td>
</tr>
<tr>
<td>VII-27.</td>
<td>Source program listing of subroutine S01 (Link 2)</td>
<td>126</td>
</tr>
<tr>
<td>VII-28.</td>
<td>Source program listing of subroutine S02 (Link 2)</td>
<td>127</td>
</tr>
<tr>
<td>VII-29.</td>
<td>Source program listing of subroutine S03 (Link 2)</td>
<td>127</td>
</tr>
<tr>
<td>VII-30.</td>
<td>Source program listing of subroutine S04 (Link 2)</td>
<td>128</td>
</tr>
<tr>
<td>VII-31.</td>
<td>Source program listing of subroutine S05 (Link 2)</td>
<td>128</td>
</tr>
<tr>
<td>VII-32.</td>
<td>Source program listing of subroutine S07 (Link 2)</td>
<td>129</td>
</tr>
<tr>
<td>VII-33.</td>
<td>Source program listing of subroutine S09 (Link 2)</td>
<td>129</td>
</tr>
<tr>
<td>VII-34.</td>
<td>Source program listing of subroutine S11 (Link 2)</td>
<td>130</td>
</tr>
<tr>
<td>VII-35.</td>
<td>Source program listing of subroutine S13 (Link 2)</td>
<td>130</td>
</tr>
<tr>
<td>VII-36.</td>
<td>Source program listing of subroutine S15 (Link 2)</td>
<td>131</td>
</tr>
<tr>
<td>VII-37.</td>
<td>Source program listing of subroutine S17 (Link 2)</td>
<td>132</td>
</tr>
<tr>
<td>VII-38.</td>
<td>Source program listing of subroutine S18 (Link 2)</td>
<td>132</td>
</tr>
<tr>
<td>VII-39.</td>
<td>Source program listing of subroutine STFS (Link 2)</td>
<td>133</td>
</tr>
<tr>
<td>VII-40.</td>
<td>Source program listing of subroutine STRA (Link 2)</td>
<td>133</td>
</tr>
<tr>
<td>VII-41.</td>
<td>Source program listing of subroutine TICK (Link 2)</td>
<td>133</td>
</tr>
<tr>
<td>VII-42.</td>
<td>Source program listing of subroutine TOPO (Link 2)</td>
<td>134</td>
</tr>
<tr>
<td>VII-43.</td>
<td>Source program listing of subroutine TRAN (Link 2)</td>
<td>134</td>
</tr>
<tr>
<td>VII-44.</td>
<td>Source program listing of subroutine TRIM (Link 2)</td>
<td>135</td>
</tr>
<tr>
<td>VII-45.</td>
<td>Source program listing of subroutine TRM (Link 2)</td>
<td>135</td>
</tr>
<tr>
<td>VII-46.</td>
<td>Source program listing of main program of Link 3 (deflection link)</td>
<td>136</td>
</tr>
<tr>
<td>VII-47.</td>
<td>Source program listing of subroutine ELST (Link 3)</td>
<td>137</td>
</tr>
<tr>
<td>VII-48.</td>
<td>Source program listing of subroutine PUNC (Link 3)</td>
<td>137</td>
</tr>
<tr>
<td>VII-49.</td>
<td>Source program listing of subroutine RESI (Link 3)</td>
<td>137</td>
</tr>
<tr>
<td>VII-50.</td>
<td>Source program listing of subroutine RESW (Link 3)</td>
<td>138</td>
</tr>
<tr>
<td>VII-51.</td>
<td>Source program listing of subroutine TICK (Link 3)</td>
<td>138</td>
</tr>
<tr>
<td>VII-52.</td>
<td>Source program listing of subroutine VELAS (Link 3)</td>
<td>139</td>
</tr>
</tbody>
</table>
## Contents (contd)

### Tables (contd)

<table>
<thead>
<tr>
<th>Page</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VII-53</td>
<td>Source program listing of main program of Link 4 (stress link) . . . . . 140</td>
</tr>
<tr>
<td>VII-54</td>
<td>Source program listing of subroutine ABEO (Link 4) . . . . . . 141</td>
</tr>
<tr>
<td>VII-55</td>
<td>Source program listing of subroutine AGEL (Link 4) . . . . . . 142</td>
</tr>
<tr>
<td>VII-56</td>
<td>Source program listing of subroutine BEST (Link 4) . . . . . . 142</td>
</tr>
<tr>
<td>VII-57</td>
<td>Source program listing of subroutine BOFI (Link 4) . . . . . . 142</td>
</tr>
<tr>
<td>VII-58</td>
<td>Source program listing of subroutine CAS4 (Link 4) . . . . . . 143</td>
</tr>
<tr>
<td>VII-59</td>
<td>Source program listing of subroutine CODI (Link 4) . . . . . . 144</td>
</tr>
<tr>
<td>VII-60</td>
<td>Source program listing of subroutine DIMI (Link 4) . . . . . . 144</td>
</tr>
<tr>
<td>VII-61</td>
<td>Source program listing of subroutine DINA (Link 4) . . . . . . 145</td>
</tr>
<tr>
<td>VII-62</td>
<td>Source program listing of subroutine EPAN (Link 4) . . . . . . 146</td>
</tr>
<tr>
<td>VII-63</td>
<td>Source program listing of subroutine FINDQ (Link 4) . . . . . . 147</td>
</tr>
<tr>
<td>VII-64</td>
<td>Source program listing of subroutine FINDX (Link 4) . . . . . . 148</td>
</tr>
<tr>
<td>VII-65</td>
<td>Source program listing of subroutine GENE (Link 4) . . . . . . 148</td>
</tr>
<tr>
<td>VII-66</td>
<td>Source program listing of subroutine INER (Link 4) . . . . . . 148</td>
</tr>
<tr>
<td>VII-67</td>
<td>Source program listing of subroutine INLZ (Link 4) . . . . . . 149</td>
</tr>
<tr>
<td>VII-68</td>
<td>Source program listing of subroutine INV (Link 4) . . . . . . 150</td>
</tr>
<tr>
<td>VII-69</td>
<td>Source program listing of subroutine LEST (Link 4) . . . . . . 150</td>
</tr>
<tr>
<td>VII-70</td>
<td>Source program listing of subroutine MDIN (Link 4) . . . . . . 151</td>
</tr>
<tr>
<td>VII-71</td>
<td>Source program listing of subroutine META (Link 4) . . . . . . 152</td>
</tr>
<tr>
<td>VII-72</td>
<td>Source program listing of subroutine QUAD (Link 4) . . . . . . 153</td>
</tr>
<tr>
<td>VII-73</td>
<td>Source program listing of subroutine REVO (Link 4) . . . . . . 154</td>
</tr>
<tr>
<td>VII-74</td>
<td>Source program listing of subroutine ROTA (Link 4) . . . . . . 155</td>
</tr>
<tr>
<td>VII-75</td>
<td>Source program listing of subroutine SAME (Link 4) . . . . . . 156</td>
</tr>
<tr>
<td>VII-76</td>
<td>Source program listing of function SCAL (Link 4) . . . . . . 156</td>
</tr>
<tr>
<td>VII-77</td>
<td>Source program listing of subroutine SETA (Link 4) . . . . . . 157</td>
</tr>
<tr>
<td>VII-78</td>
<td>Source program listing of subroutine STRA (Link 4) . . . . . . 158</td>
</tr>
<tr>
<td>VII-79</td>
<td>Source program listing of subroutine STRS (Link 4) . . . . . . 158</td>
</tr>
<tr>
<td>VII-80</td>
<td>Source program listing of subroutine TEMP (Link 4) . . . . . . 159</td>
</tr>
<tr>
<td>VII-81</td>
<td>Source program listing of subroutine TICK (Link 4) . . . . . . 159</td>
</tr>
<tr>
<td>VII-82</td>
<td>Source program listing of subroutine TOPO (Link 4) . . . . . . 159</td>
</tr>
<tr>
<td>VII-83</td>
<td>Source program listing of subroutine TRAN (Link 4) . . . . . . 160</td>
</tr>
<tr>
<td>VII-84</td>
<td>Source program listing of subroutine UNIT (Link 4) . . . . . . 160</td>
</tr>
<tr>
<td>VII-85</td>
<td>Source program listing of subroutine VECT (Link 4) . . . . . . 160</td>
</tr>
</tbody>
</table>
Contents (contd)

Figures

VI-1. Flowchart of main program of Link 1 (input link) .......................... 27
VI-2. Flowchart of subroutine ARAN (Link 1) ........................................ 29
VI-3. Flowchart of subroutine COOR (Link 1) ......................................... 31
VI-4. Flowchart of subroutine EXCH (Link 1) ........................................... 32
VI-5. Flowchart of function LEBIN (Link 1) ............................................ 32
VI-6. Flowchart of subroutine SEBIN (Link 1) ......................................... 33
VI-7. Flowchart of subroutine MEST (Link 1) ........................................... 34
VI-8. Flowchart of subroutine OUTPT (Link 1) ........................................ 36
VI-9. Flowchart of subroutine SRAT (Link 1) ........................................... 37
VI-10. Flowchart of subroutine TABL (Link 1) ......................................... 38
VI-11. Flowchart of subroutine TICK (Link 1) ......................................... 39
VI-12. Flowchart of subroutine TOPO (Link 1) ........................................ 40
VI-13. Flowchart of main program of Link 2 (generation link) .................... 42
VI-14. Flowchart of subroutine ADM (Link 2) .......................................... 45
VI-15. Flowchart of subroutine BEAM (Link 2) ......................................... 46
VI-16. Flowchart of subroutine CODI (Link 2) ......................................... 47
VI-17. Flowchart of subroutine CORT (Link 2) ........................................ 48
VI-18. Flowchart of subroutine CUTE (Link 2) ......................................... 49
VI-19. Flowchart of subroutine DARN (Link 2) ....................................... 50
VI-20. Flowchart of subroutine DMM (Link 2) ......................................... 51
VI-21. Flowchart of subroutine ELDI (Link 2) ......................................... 51
VI-22. Flowchart of subroutine PLBE (Link 2) ......................................... 52
VI-23. Flowchart of subroutine RLOC (Link 2) ....................................... 52
VI-24. Flowchart of subroutine S01 (Link 2) .......................................... 53
VI-25. Flowchart of subroutine S02 (Link 2) .......................................... 54
VI-26. Flowchart of subroutine S03 (Link 2) .......................................... 55
VI-27. Flowchart of subroutine S04 (Link 2) .......................................... 56
VI-28. Flowchart of subroutine S05 (Link 2) .......................................... 57
VI-29. Flowchart of subroutine S07 (Link 2) .......................................... 58
VI-30. Flowchart of subroutine S09 (Link 2) .......................................... 59
VI-31. Flowchart of subroutine S11 (Link 2) .......................................... 60
VI-32. Flowchart of subroutine S13 (Link 2) .......................................... 60
Contents (contd)

Figures (contd)

VI-33. Flowchart of subroutine S15 (Link 2) ........................................ 61
VI-34. Flowchart of subroutine S17 (Link 2) ........................................ 64
VI-35. Flowchart of subroutine S18 (Link 2) ........................................ 66
VI-36. Flowchart of subroutine STFS (Link 2) ........................................ 67
VI-37. Flowchart of subroutine STRA (Link 2) ........................................ 68
VI-38. Flowchart of subroutine TOPO (Link 2) ........................................ 69
VI-39. Flowchart of subroutine TRAN (Link 2) ........................................ 70
VI-40. Flowchart of subroutine TRIM (Link 2) ........................................ 70
VI-41. Flowchart of subroutine TRM (Link 2) ........................................ 70
VI-42. Flowchart of main program of Link 3 (deflection link) ..................... 71
VI-43. Flowchart of subroutine ELST (Link 3) ........................................ 72
VI-44. Flowchart of subroutine RESI (Link 3) ........................................ 74
VI-45. Flowchart of subroutine RESW (Link 3) ........................................ 75
VI-46. Flowchart of subroutine VELAS (Link 3) ........................................ 76
VI-47. Flowchart of main program of Link 4 (stress link) .......................... 78
VI-48. Flowchart of subroutine ABEQ (Link 4) ........................................ 81
VI-49. Flowchart of subroutine BEST (Link 4) ........................................ 82
VI-50. Flowchart of subroutine BOFI (Link 4) ........................................ 83
VI-51. Flowchart of subroutine DIMI (Link 4) ........................................ 88
VI-52. Flowchart of subroutine DINA (Link 4) ........................................ 89
VI-53. Flowchart of subroutine EPAN (Link 4) ........................................ 90
VI-54. Flowchart of subroutine FINDQ (Link 4) ...................................... 92
VI-55. Flowchart of subroutine FINDX (Link 4) ...................................... 92
VI-56. Flowchart of subroutine GENE (Link 4) ........................................ 92
VI-57. Flowchart of subroutine INER (Link 4) ........................................ 94
VI-58. Flowchart of subroutine INLZ (Link 4) ........................................ 94
VI-59. Flowchart of subroutine INV (Link 4) ........................................ 95
VI-60. Flowchart of subroutine LEST (Link 4) ........................................ 97
VI-61. Flowchart of subroutine Mدين (Link 4) ........................................ 98
VI-62. Flowchart of subroutine META (Link 4) ....................................... 99
VI-63. Flowchart of subroutine QUAD (Link 4) ....................................... 101
VI-64. Flowchart of subroutine REVO (Link 4) ....................................... 103
Contents (contd)

Figures (contd)

VI-65. Flowchart of subroutine ROTA (Link 4) ........................................... 105
VI-66. Flowchart of subroutine SAME (Link 4) ........................................... 106
VI-67. Flowchart of function SCAL (Link 4) .................................................. 107
VI-68. Flowchart of subroutine SETA (Link 4) ............................................. 107
VI-69. Flowchart of subroutine STRS (Link 4) ............................................. 110
VI-70. Flowchart of subroutine TEMP (Link 4) ............................................. 111
VI-71. Flowchart of subroutine UNIT (Link 4) ............................................ 111
VI-72. Flowchart of subroutine VECT (Link 4) ............................................. 111
Abstract

A general-purpose digital computer program (named ELAS) for the in-core solution of linear equilibrium problems of structural mechanics is described for potential and actual users in Volume I of this report, and is documented in Volume II. The program requires minimum input for the description of the problem. The solution is obtained by means of the displacement method and the finite element technique. Almost any geometry and structure may be handled because of the availability of lineal, triangular, quadrilateral, tetrahedral, hexahedral, conical, triangular torus, and quadrilateral torus elements. The assumption of piecewise linear deflection distribution insures monotonic convergence of the deflections from the stiffer side with decreasing mesh size. The stresses are provided by the best-fit strain tensors in the least-squares sense at the mesh points where the deflections are given. The selection of local coordinate systems whenever necessary is automatic. The core memory is efficiently used by means of dynamic memory allocation, an optional mesh-point relabelling scheme, imposition of the boundary conditions during the assembly time, and the straight-line storage of the rows of the stiffness matrix within variable bandwidth and the main diagonal. The number of unsuppressed degrees of freedom that can be handled in a given problem is 500 to 600 for a typical structure, but might far exceed these average values for special types of problems; the execution time of such problems is about four minutes in 32K IBM 7094 Model I machines. The program is written in FORTRAN II language. The source deck consists of about 8000 cards and the object deck contains about 1400 binary cards. The physical program (standard ELAS) is available from COSMIC, the agency for the distribution of NASA computer programs.
I. Introduction

Volume I, *User’s Manual*, of this report gives a general description of ELAS,* a general-purpose digital computer program for the in-core solution of linear equilibrium problems of structural mechanics, and contains the information necessary for input preparation, arrangement of the physical program, and interpretation of output and error messages. Volume II, *Documentation of the Program*, is published in two parts: the basic Volume II, which gives the theoretical background of the program and contains tables and figures describing the COMMON variables, their meanings, and their arrangement in COMMON; and this report—Addendum to Volume II—which contains program descriptions, flowcharts, and source program listings for all program elements of ELAS/Level 3. (The original version of the ELAS program made available from COSMIC** in April 1968 is designated ELAS/Level 0. Subsequent program corrections made in January 1969, March 1969, and May 1969 updated the program to ELAS/Level 1, ELAS/Level 2, and ELAS/Level 3, respectively.)

Sections II, III, IV, and V of the Addendum briefly describe the main programs and the subroutines of Links 1, 2, 3, and 4, respectively, of the ELAS program with reference to the flowcharts illustrated in Section VI, and the source program listings given in Section VII. Program descriptions include all subroutines that are not in the FORTRAN library. The standard ELAS is coded in FORTRAN II, with the exception of subroutines LEBIN, SEBIN, and TICK, which are in FAP. The subroutines are described in alphabetical order under each main program. The flowcharts, which are also in alphabetical order, present semidetailed diagrams of the sequential logic and decision points in the program. The source program listings are a straight listing of the first file in the program tape that contains the physical program.

The user of this Addendum will need both Volume I and the basic Volume II for reference because of numerous cross-references to figures and tables contained therein. The information in the referenced figures and tables is essential to interpretation of the content of the Addendum. Reference is also made herein to program input and output items and error messages, which are described and identified by number in Volume I.

---

*First two syllables of the word Elasticity.

**Computer Software Management and Information Center, Computer Center, University of Georgia, Athens, Georgia, 30601, telephone 404-542-3265.
II. Main Program and Subroutines of Link 1

A. Main Program of Link 1

The flowchart and the source program listing of the main program of Link 1 are given in Fig. VI-1 and Table VII-1, respectively. The logical function of the program may be summarized as follows:

1. It defines IN, IT, IDEG, ITYPE, IGEFL, ISTR, IH, IBN, IP, IPR, IMAT, NTIC, ISDT, ISDY, ISDZ, IARE, IMM, IMZ, IMFI, INX, INP, ISHUF, ICOR, IBUN, IMES, IPR, ITAP, ITAS, G1, G2, G3, ACEL directly from Input Item 2.

2. It computes constants ISUM, IND, IORD, IORD1, and ZGEM, and pointers J1, J2, J3, J4, J5, J6, J7, J8, J9, J10, IBB, IBO, ID, IDT, IDY, ISE, ICAR, ICX, ICY, ICZ, ICFL, IXX, IYY, IZZ, IIC, IDEF, IST, IIS, IU, and IDZ from the information given by Input Item 2.

3. It generates the vectors related with pointers J1, J2, J3, J4, J5, J6, J7, J8, J9, J10, IBB, IBO, IIC, IXX, IYY, IZZ, and IDEFL (partially) directly or indirectly from the input information of the job.

4. Depending upon the value of INX, it transfers control either to statement 2700 or to the main program of Link 3.

Before using any input information, the main program checks it against the input specifications (see Sect. III and IV, Vol. I). If the program encounters an irrecoverable error in the input information, it always branches to statement 300, which prints out COMMON both in fixed- and floating-point modes and skips the related job. In transferring information from input cards into the proper locations in COMMON, the program uses DUMMY (also called IDUM) area in COMMON for temporary storage. The main program calls subroutine TABL to print out Output Item 1; subroutine TICK to measure time; subroutine BUNG to generate deflection boundary conditions (dbc) input units automatically, if IBUN = 1; subroutine CORG to generate coordinates of the nodes automatically, if ICOR = 1; subroutine MESG to generate mesh topology and element properties automatically, if IMES = 1; subroutine COOR to read in, examine, and store nodal coordinates; subroutine MEST to read in and store element data; subroutine TOPO to examine and separate the element data in storage; and finally, subroutine SRAT to obtain internal node labels in vector ISIR, and the highest internal label in the node set of each node in vector IMAX (refer to Input Item 17, Sect. IV, Vol. I). Among these subroutines, subroutine SRAT has its own subroutine. The contribution of the prescribed concentrated loads to the right-hand-side vector of the equilibrium equations before the displacement boundary conditions are imposed is stored in the
IDENT-pointer-related vector, first as in item (1) of the IDENT entry of Table III-3, Vol. II (basic), and then as in item (2) in the same entry. Probably the most important function of the main program is the generation of vectors defined by pointers IBB, IBO, and IIC. The meanings of the entries of these vectors are given in Table III-3, Vol. II (basic), and Table VI-2, Vol. I. These vectors are first generated as if all deflections were independent and with IBB numbers always equal to IND + 1. Then the numbers are modified with the ddc input units to recognize linear dependence. Finally, when vectors ISIR and IMAX are provided by subroutine SRAT, they are finalized. Vectors defined by pointers IBB, IBO, and IIC are first used in the main program to compute the contributions of the prescribed concentrated loads to the reduced right-hand-side vector in the IDENT-pointer-related vector. Later in Link 2, they are used in computing the contributions of the element stiffness matrices to the reduced stiffness matrix in the IDENT-pointer-related vector and the reduced load vector in the IDENT-pointer-related vector, and the contributions of the element load vectors to the reduced load vector in the IDENT-pointer-related vector. In Link 3, these three vectors are used in obtaining the deflections of all nodes from the reduced deflection vector in the IDENT-pointer-related vector. The standard ELAS Link 1 main program assumes that IN ≤ 540 and ISUM < 10000. (See Appendix, Vol. I, for instructions on how to change these limits.)

B. Subroutines of Link 1

1. Subroutine ARAN. Subroutine ARAN is called by subroutine SRAT. The flowchart and the source program listing of ARAN are given in Fig. VI-2 and Table VII-2, respectively. The logical function of the subroutine may be summarized as follows:

(1) Subroutine ARAN generates vector IMAX for subroutine SRAT.

(2) If 0 < ISHUF < 3, the subroutine modifies vector ISIR and computes vector IMAX accordingly, to minimize the shaded area of the coefficient matrix shown in Fig. II-1 (Vol. I).

(3) If ISHUF = 2, the subroutine reads cards for vector ISIR, modifies matrix ABIN accordingly, and performs the function given in (2).

(4) The subroutine produces relabelling output items (Output Item 5) according to Sect. VI-F, Vol. I.

In performing these tasks, subroutine ARAN expects that connectivity matrix ABIN, ISIR vector of labels, row order IN, and column order ISUR of matrix ABIN are available in COMMON. In performing logical function (2), the subroutine also generates vector IMLN. Subroutine ARAN calls subroutine OUTPT to print out mesh topology (P) of Output Item 5 (see Sect. VI-F, Vol. I), subroutine EXCH to interchange to successive rows and their respective columns in the connectivity matrix ABIN, function LEBIN to find out if a node is connected with another node, and subroutine TICK to measure relabelling time. The algorithm for logical function (2) is given in Ref. 1. The standard ELAS assumes that a word consists of 36 binary bits, and this is assumed in subroutine ARAN. (See Appendix, Vol. I, for instructions on how to change this constraint.)

2. Subroutine BUNG. Subroutine BUNG is called by the main program of Link 1, if IBUN = 1. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-3. If IBUN = 1, the subroutine should be rewritten by the user, as explained in Sect. V-D, Vol. I. The logical function of the subroutine is to place IBN number of ddc input units into DUMMY or IDUM area.

3. Subroutine COOR. Subroutine COOR is called by

(1) Subroutine ARAN generates vector IMAX for subroutine SRAT.

(2) If 0 < ISHUF < 3, the subroutine modifies vector ISIR and computes vector IMAX accordingly, to minimize the shaded area of the coefficient matrix shown in Fig. II-1 (Vol. I).

(3) If ISHUF = 2, the subroutine reads cards for vector ISIR, modifies matrix ABIN accordingly, and performs the function given in (2).

(4) The subroutine produces relabelling output items (Output Item 5) according to Sect. VI-F, Vol. I.

In performing these tasks, subroutine ARAN expects that connectivity matrix ABIN, ISIR vector of labels, row order IN, and column order ISUR of matrix ABIN are available in COMMON. In performing logical function (2), the subroutine also generates vector IMLN. Subroutine ARAN calls subroutine OUTPT to print out mesh topology (P) of Output Item 5 (see Sect. VI-F, Vol. I), subroutine EXCH to interchange to successive rows and their respective columns in the connectivity matrix ABIN, function LEBIN to find out if a node is connected with another node, and subroutine TICK to measure relabelling time. The algorithm for logical function (2) is given in Ref. 1. The standard ELAS assumes that a word consists of 36 binary bits, and this is assumed in subroutine ARAN. (See Appendix, Vol. I, for instructions on how to change this constraint.)

2. Subroutine BUNG. Subroutine BUNG is called by the main program of Link 1, if IBUN = 1. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-3. If IBUN = 1, the subroutine should be rewritten by the user, as explained in Sect. V-D, Vol. I. The logical function of the subroutine is to place IBN number of ddc input units into DUMMY or IDUM area.

3. Subroutine COOR. Subroutine COOR is called by

(1) Subroutine ARAN generates vector IMAX for subroutine SRAT.

(2) If 0 < ISHUF < 3, the subroutine modifies vector ISIR and computes vector IMAX accordingly, to minimize the shaded area of the coefficient matrix shown in Fig. II-1 (Vol. I).

(3) If ISHUF = 2, the subroutine reads cards for vector ISIR, modifies matrix ABIN accordingly, and performs the function given in (2).

(4) The subroutine produces relabelling output items (Output Item 5) according to Sect. VI-F, Vol. I.

In performing these tasks, subroutine ARAN expects that connectivity matrix ABIN, ISIR vector of labels, row order IN, and column order ISUR of matrix ABIN are available in COMMON. In performing logical function (2), the subroutine also generates vector IMLN. Subroutine ARAN calls subroutine OUTPT to print out mesh topology (P) of Output Item 5 (see Sect. VI-F, Vol. I), subroutine EXCH to interchange to successive rows and their respective columns in the connectivity matrix ABIN, function LEBIN to find out if a node is connected with another node, and subroutine TICK to measure relabelling time. The algorithm for logical function (2) is given in Ref. 1. The standard ELAS assumes that a word consists of 36 binary bits, and this is assumed in subroutine ARAN. (See Appendix, Vol. I, for instructions on how to change this constraint.)

4. Subroutine CORG. Subroutine CORG is called by the main program of Link 1, if ICOR = 1. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-5. If ICOR = 1, the subroutine should be rewritten by the user, as explained in Sect. V-D, Vol. I. The logical function of this subroutine, when ICOR = 1, is to generate vectors related with pointers IXX, IYY, and IZZ.

5. Subroutine EXCH. Subroutine EXCH is called by subroutine ARAN. The flowchart and the source program listing of EXCH are given in Fig. VI-4 and Table VII-6, respectively. The function of this subroutine is to read the cards of Input Item 14 in blocks of not greater than 1000 nodes, to examine whether the node labels are sequential, and to generate IXX-, IYY-, and IZZ-pointer-related vectors. If IGEM = 0, the IZZ-pointer-related vector will not be generated. In case of error, the subroutine returns control to the calling program with IERR = 1, without completing its function.
of the same matrix, MI is the smallest binary column number of the first nonzero binary entry either in row I or in row IP; MX is the largest binary column number of the last nonzero binary entry either in row I or in row IP. Since ABIN is a symmetric binary matrix, MI and MX also define the limits of the Ith and IPth columns. Interchange operation is carried out only within these limits. The subroutine expects ABIN, I, IP, MI, and MX to be available in COMMON, and it assumes that a word consists of 36 binary bits. (See Appendix, Vol. I for instructions on how to change this limit.) Subroutine EXCH calls function LEBIN to obtain the value of a certain bit of a word, and subroutine SEBIN to store 1 or 0 in a certain bit of a word.

6. Function LEBIN. The subprogram LEBIN is called by subroutines ARAN and EXCH. The flowchart and the source program listing of LEBIN are given in Fig. VI-5 and Table VII-7, respectively. The coding is in FAP for the IBM 7094. The logical function of LEBIN is to return as a FORTRAN integer the value of the bit, shown in the second argument, of the word shown in the first argument. It assumes that the word length is 36 binary bits. For any other machine, this function should be rewritten.

7. Subroutine SEBIN. Subroutine SEBIN is called by subroutines EXCH and SRAT. The flowchart and the source program listing of SEBIN are given in Fig. VI-6 and Table VII-7, respectively. The coding is in FAP for the IBM 7094. The logical function of SEBIN is to store 1 or 0 (as shown in the third argument) in to the bit (shown in the second argument) of the word (shown in the first argument). The subroutine assumes that a word consists of 36 binary bits. For any other machine, this subroutine should be rewritten.

8. Subroutine MESS. Subroutine MESS is called by the main program of Link 1, if IMES = 1. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-8. If IMES = 1, the subroutine should be rewritten by the user, as explained in Sect. V-C, Vol. I. The logical function of this subroutine, when IMES = 1, is to generate vectors related with pointers J1, J2, J3, J4, J5, J6, J7, J8, J9, and J10.

9. Subroutine MEST. Subroutine MEST is called by the main program of Link 1, if IMES = 0. The flowchart and the source program listing of MEST are given in Fig. VI-7 and Table VII-9, respectively. The function of this subroutine is first to read the cards of Input Item 16, one or more at a time, into DUMMY area, and to check the validity of M number of the element descriptors (see Table IV-3, Vol. I), then to store words J1W, J2W, J3W, J4W, J5W, and if they exist, J6W, J7W, J8W, J9W, J10W of the element descriptors into the proper locations in the respective vectors related with pointers J1, J2, J3, J4, J5, J6, J7, J8, J9, and J10, and to check the positivity of the vertex labels. In case of error, the subroutine returns control to the calling program with IERR = 1, without completing its function. If no error is encountered, the subroutine performs its operations on the input cards of every element, sequentially, until all element data are processed.

10. Subroutine OUTPT. Subroutine OUTPT is called by subroutine ARAN, if INP = 2. The flowchart and the source program listing of OUTPT are given in Fig. VI-8 and Table VII-10, respectively. The function of this subroutine is to print out Output Item 5(P) (see Sect. VI-F, Vol. I). This subroutine assumes that a word is 36 binary bits and IN ≤ 540. (See Appendix, Vol. I for instructions on how to change these limits.) Subroutine OUTPT expects ISIR, IMIN, IMAX vectors, ABIN matrix, IN, and ISUR to be available in COMMON.

11. Subroutine SRAT. Subroutine SRAT is called by the main program of Link 1. The flowchart and the source program listing of SRAT are given in Fig. VI-9 and Table VII-11, respectively. The functions of this program may be summarized as follows:

(1) If ISHUF = 3, subroutine SRAT reads the cards of Input Item 17 for the generation of internal node labels in vector ISIR, and the highest internal label in the node set of each node in vector IMAX (refer to Input Item 17, Sect. IV, Vol. I). If ISHUF ≠ 3, subroutine SRAT generates vector ISIR by assuming that internal and external labels of the nodes are the same; computes ISUR, which is the number of words whose binary bits are enough to provide one bit for each of the IN nodes; generates connectivity matrix ABIN from the mesh-topology information of Input Item 16 and from the deflection boundary condition input units already in the IBO-pointer-related vector; and calls subroutine ARAN to obtain the finalized values of vectors ISIR and IMAX. (In generating the binary connectivity matrix ABIN, subroutine SRAT first clears matrix ABIN; then it processes elements one at a time by first listing the labels of the vertices of the element and the labels of the nodes connected with these vertices through the dbc.
input items; and then indicates, by means of a binary 1 in the proper places in matrix ABIN, the fact that all these nodes are connected with each other. In matrix ABIN, one row and one column are assigned to every node so that ABIN is a symmetric matrix. If a node \( i \) is connected with a node \( j \), the \( i \)th row and the \( j \)th column, and likewise the \( j \)th row and the \( i \)th column, of the binary matrix ABIN contain a binary bit 1. All diagonal elements of the binary matrix ABIN are binary bit 1. If a node is constrained completely by means of dbc input items the only nonzero bit in the row and the column of this node is on the diagonal. The ordering of rows and columns of the binary matrix ABIN is done by the order that is available in vector ISIR.)

(2) If \( INP \neq 0 \), subroutine SRAT generates and prints out Output Item 6.

(3) It generates the vector related with pointer IU, and if \( INP \neq 0 \), prints it as Output Item 9.

(4) It computes the number of words between points D and F, and between the beginning of COMMON and point G, designated in Fig. III-1, Vol. II (basic), and prints out Output Items 7 and 8.

(5) It prints Error Message 5 (see Table VII-1, Vol. I) if the number of words to the left of point G is less than 12,750, which is based on the assumption that the core memory is 32,768 words. (See Appendix, Vol. I for instructions on how to modify the program for other core memory sizes.)

Subroutine SRAT calls subroutine TOPO in obtaining the vertex labels of the elements, subroutine SEBIN in generating binary matrix ABIN, and subroutine ARAN to obtain vector IMAX and the corresponding vector ISIR.

12. Subroutine TABL. Subroutine TABL is called by the main program of Link 1. The flowchart and the source program listing of TABL are given in Fig. VI-10 and Table VII-12, respectively. The function of this subroutine is to print out Output Item 1. It expects the contents of Input Items 1 and 2 to be available in COMMON.

13. Subroutine TICK. Subroutine TICK is called by the main program of Link 1. It is also called by subroutine ARAN and by the main programs of Links 2, 3, and 4. The flowchart and the source program listing of subroutine TICK are given in Fig. VI-11 and Table VII-13, respectively. The coding is in FAP for the IBM 7094. The subroutine expects the time in units of 1/60 second in the absolute memory location 5 as a binary integer. Its function is to return, as a FORTRAN integer in the argument, the time, in 1/60-second units, elapsed since the first call. (See Appendix, Vol. I, for instructions on how to change this subroutine for other machines.)

14. Subroutine TOPO. Subroutine TOPO is called by the main program of Link 1. It is also called by subroutine SRAT. The flowchart and the source program listing of TOPO are given in Fig. VI-12 and Table VII-14, respectively. The function of this subroutine may be summarized as follows:

(1) It extracts from J1-, J2-, J3-, J4-, J5-, J6-, J7-, J8-, J9-, and J10-pointer-related vectors for the \( Mth \) element the vertex labels in \( N_i \) vector, the pressure type number in JPRS, the material type number in IMET, the thickness type number in ITIC, the temperature increase type number in ITEM, the temperature gradient in \( y \)-direction type number in JSDY, the temperature gradient in \( z \)-direction type number in JSDZ, the cross-sectional area type number in JARE, the torsional constant type number in JMMX, the \( y \)-moment of inertia type number in JMMY, the \( z \)-moment of inertia type number in JMMZ, and the angle for principal axes type number in JMFI, as described in Table III-2 (Vol. II, basic).

(2) It checks whether the vertex labels and the property type numbers are within the bounds prescribed in Input Item 2. In case of error, the subroutine continues scanning the properties of the \( Mth \) element and prints out Error Message 3 (see Table VII-1, Vol. I) after executing the implementation of the message.
III. Main Program and Subroutines of Link 2

A. Main Program of Link 2

The flowchart and the source program listing of the main program of Link 2 are given in Fig. VI-13 and Table VII-15, respectively. The logical function of the program may be summarized as follows:

1. It clears the reduced stiffness matrix area (the IST-pointer-related vector of Table III-3, Vol. II, basic).

2. It generates the elemental matrices in S and P areas, and assembles these into IST- and IDEF-pointer-related vectors, sequentially.

3. It stores on tape ITAS (if available) the elemental matrices, sequentially.

4. Depending upon the value of INX, it transfers control either to the main program of Link 1 or to the main program of Link 3.

In carrying out function (2) listed above, the program executes a DO-loop on element labels M. In this loop, for any element M, it first clears a certain work area (see block *133 in Fig. VI-13) and sets the variables ITTT = 0, ITTM = NAV = 1, and CFE = 1. Variable ITTM is not in COMMON. For element types 1, 2, 3, 4, 5, 7, 9, 11, 13, 15, 17, and 18, ITTM = NAV = 1 and CFE = 1., and ITTT is made 1 at block 5100 of Fig. VI-13. For the remaining element types, that is, element types 6, 8, 10, 12, 14, and 16, the program establishes subelements as described in Table VI-5 (Vol. I). For element types 6, 8, 12, 14, and 16, the program obtains two triangles for every quadrilateral in two ways, as shown in Table VI-5 (Vol. I). Since such a procedure is equivalent to doubling the material volume of the structure, the elemental matrices are weighted with constant CFE = 1/2 (See block 4902 of Fig. VI-13); ITTM is the number of the subelements and ITTT is the subelement count in each way of subdivision, and NAV = 1 is the count of subdivisions. For example, NAV = 2, ITTT = 2 means the second triangle of the first way of subdivision, and NAV = 3, ITTT = 1 means the first triangle of the second way of subdivision. The same symbolism applies for element type 10, where ITTM = 5, which indicates that there are five tetrahedrons for every way of subdividing an hexahedron. The subdivision procedure is achieved as indicated in block 504 of Fig. VI-13, with the help of subroutine CUTE.
In the DO-loop on elements, after the initialization of block *133 (Fig. IV-13), by means of subroutine TOPO, the descriptive words of the element (the quantities listed in J1- through J10-pointer-related vectors) are extracted and analyzed to obtain the vertex labels in N block and the property type numbers. Next, the vertex labels are copied to NOO block in preparation for the subdivision operation and IMS is established. Following this, the order of the element stiffness matrix IDS is determined, actual values of load and geometry constraints are obtained, and the material constants are prepared. Even if an element is subdivided, the same load, geometry, and material constants are used for the subelements. The following constants are prepared, as explained in Sect. III-C, Vol. I: E and G for element types 1, 2, 3, and 4; D21 for element types 9, 10, 15, and 16; D33 for element types 5, 6, 7, 8, 11, 12, 13, and 14; and E22 for element types 7, 8, 11, 12, and 18. The constant E is the Young's modulus, G is the shear modulus, D21 is the upper half of the 6x6 material matrix, D33 is the material constants matrix for in-plane deformations, and E22 is the material constants matrix for out-of-plane deformations. Finally, arrays related with subelement vertex coordinates and labels are prepared (see block 5100, Fig. VI-13), the subelement count ITTT is set, and the number of entries in the free-free stiffness matrix, IDS2, is obtained; and S and P areas are cleared for elemental matrices of the subelement/element (see block 5600, Fig. VI-13).

In the DO-loop on elements, for every subelement/element, a free-free stiffness matrix and a load vector are generated in S and P, respectively. For this purpose the program calls subroutine STFS, which, in turn, calls the proper subroutine determined by the type number of the current element. These subroutines are S01, S02, S03, S04, S05, S07, S09, S11, S13, S15, S17, and S18. The numeral in these names corresponds to the type number of the element for which the subroutine is directly applicable. In all these subroutines, the input information is in X, Y, Z, XD, YD, ZD, DT, DG, DGY, DGZ, TE, AL1, AL2, AL3, E, G, D21, D33, E22, PRES, ACEL, N, CONS, UV, and COMMON 200-328 locations. The output consists of S, P, and sometimes IPBG and IPEN constants. The latter two constants indicate whether the element load vector in P is complete, or whether some additional operation is necessary in the main program. If IPBG is nonpositive and IPBC<IPEN, no additional operation is expected from the main program to modify load vector P. The thermal portion of load vector P is always completed by the main program. Before the subroutines are called, the main program sets the list of vertex deflections due to DT of the free-free element. The subroutines called by subroutine STFS modify this vector properly so that the portion of the element load vector due to temperature changes can be added to P as the product of the free-free stiffness matrix of the element (the quantities in S) times the deflections in UV by means of subroutine DMM. This is shown in block 951 of Fig. VI-13. In this figure, Block 953 corresponds to the inquiries on constants IPEN and IPBC. The modification to vector P in the main program consists of adding certain constant values derived from CONS, PRBC, and PD values to certain subvectors of P as indicated by IPEN and IPBC. The values of IPEN, IPBC, CONS, PRBC, and PD are determined by the subroutine that generates S and P.

After the generation of S and P in the DO-loop on elements, the main program scans each entry of S and P one at a time and assembles it to the governing equations of the system. This is the operation in block 9332 of Fig. VI-13, which ends just before block 95*. The assembly procedure is described in Sect. II, Vol. II (basic). With the notation given there, a, e, , e, , and b, e, , are generated in IQE, CCCL, IBS and JQE, CCCJ, JBS locations of COMMON, respectively, by means of subroutine DARN. After the assembly of the subelement/element matrices S and P, if a scratch tape is available, the operations shown in block 9982 (Fig. IV-13) are for future reference. Next, ITTT is compared with ITTM, and the value of NAV (which is updated by subroutine CUTE) is inquired. When the last subelement of the last subdivision is completed, the process is repeated for the next element until all the elements are handled. Then, scratch tape ITAS (if prescribed) is rewound, the total time elapsed since the first entry to the main program is obtained by means of subroutine TICK and recorded, and the transfer of control is made.

The main program of Link 2 is also responsible for the production of Output Items 14, 15, 16, and 17, as prescribed by INP and subroutine CAS2 (see Table VI-1, Vol. I). At the end of Link 2, the coefficients in the shaded areas of Fig. II-1, Vol. I, are generated and stored in IST- and IDEF-pointer-related arrays of COMMON.

### B. Subroutines of Link 2

1. **Subroutine ADM.** Subroutine ADM is called by subroutines S05, S07, S09, S15, S17, and S18. The flowchart and the source program listing of ADM are given...
in Fig. VI-14 and Table VII-16, respectively. The subroutine has seven arguments. The function of the subroutine is to add a constant times a square matrix to another one which is symmetric. The constant is given by the seventh argument, the matrix to be added is given by the third argument, and the matrix to be increased is given by the first argument. The order of the latter matrix is given in the second argument, the order of the former by the fourth argument. Since the orders are different, the row and column numbers of the entry in the matrix of the first argument corresponding to the first entry of the matrix of the third argument are given with the fifth and sixth arguments. The addition operation is carried out such that the matrix of the first argument always remains symmetrical. Both matrices are assumed to be listed columnwise, with the column orders as prescribed by the second and fourth arguments. The order of the matrix in the third argument cannot be larger than 4. There is no error return of the subroutine. In all cases, the matrix in the first argument is the free-free stiffness matrix of various types of elements, and the matrix in the third argument is usually a submatrix related with given degree-of-freedom directions.

2. Subroutine BEAM. Subroutine BEAM is called by subroutines S02 and S04. The flowchart and the source program listing of BEAM are given in Fig. VI-15 and Table VII-17, respectively. This subroutine generates, in local coordinates, the free-free stiffness matrix of a planar beam element and stores it in A(6,6), which is located in COMMON (200). The matrix may be partitioned with respect to degrees of freedom (nine submatrices, 2 x 2 each). In generation of the stiffness matrix, the shear deformations are ignored.

3. Subroutine CAS2. Subroutine CAS2 is called by the main program of Link 2. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-18. If Output Items 14, 15, and 16 are to be produced selectively, the subroutine should be rewritten by the user, as explained in Sect. V-G, Vol. I. The logical function of this subroutine is to change the value of INP as desired in the DO-loop on elements of the main program of Link 2.

4. Subroutine CODI. Subroutine CODI is called by subroutines S02, S03, and S04. The flowchart and the source program listing of CODI are given in Fig. VI-16 and Table VII-19, respectively. The subroutine generates the direction cosines of the local axes for element types 2, 3, and 4 in DIR(3,3) array, which is located in COMMON(264). The first row of DIR corresponds to local x-axis, the second row corresponds to local y-axis, and the third row corresponds to local z-axis. (See Table III-3, Vol. I, and the description of Input Item 13, Sect. IV-B, Vol. I, for the rules covering the local coordinate systems of these elements.) The subroutine sets IERR = 1 and returns control to the calling program as soon as an error is detected.

5. Subroutine CORT. Subroutine CORT is called by subroutines S11 and S13. The flowchart and the source program listing of CORT are given in Fig. VI-17 and Table VII-20, respectively. The subroutine generates the direction cosines of the local coordinate axes for elements 11 and 13 in DIR(3,3) array, which is located in COMMON(264). The first, second, and third rows of DIR correspond to the first, second, and third local axis, respectively. (See Table III-3, Vol. I, for the rules in selecting the local coordinate system for elements 11 and 13.) After computation of direction cosines, the subroutine replaces X, Y, and Z values with the coordinates of the vertices in a coordinate system located usually at the centroid of the element and yet parallel to the local coordinate system of the element. Next, the subroutine computes in XD, YD, and ZD the coordinates of the second and third vertex in a coordinate system that is located at the first vertex, yet is parallel to the local coordinate system of the element. There is no error return in the subroutine.

6. Subroutine CUTE. Subroutine CUTE is called by the main program of Link 2. The flowchart and the source program listing of the subroutine are given in Fig. VI-18 and Table VII-21, respectively. This subroutine has the ITTM value as an argument (see Sect. III-A). The subroutine is called twice for element types 6, 8, 10, 12, 14, and 16 and is not called for other types of elements. Each time it is called for an element, the subroutine increments NAV (see Sect. III-A) by 1; determines ITTM, IMS, IELT, IDS values for the sub-elements; and generates in NOO array the list of mesh-point labels that conform to the (NAV − 1)st row in part A or B of Table VI-5, Vol. I, depending upon whether the value of ITTM is 2 or 5, respectively. For example, for a quadrilateral element with mesh-point labels 13, 8, 51, 16, the value of ITTM is 2, and if NAV = 2, according to the first line of Table VI-5A, Vol. I, the NOO array contains the following list: 13, 8, 51, 16, 13. There is no error return in the subroutine.

7. Subroutine DARN. Subroutine DARN is called by the main program of Link 2. The flowchart and the
source program listing of the subroutine are given in Fig. VI-19 and Table VII-22, respectively. The subroutine has four arguments. The last three arguments, KBS, CCC, and KQE, correspond to $i$, $Z$, and $a$ (or $j$, $e$, and $b$), respectively, of Sect. II, Vol. II (basic). The first argument is the label of the degree-of-freedom direction under question (see Sect. III-A). To achieve this function, subroutine DARN interprets the entries of IBB-, IBO-, and IIC-pointer-related vectors, as described in Table VI-2, Vol. I. In case of error, the last argument is set to zero, and the subroutine returns control to the calling program.

8. Subroutine DMM. Subroutine DMM is called by the main program of Link 2. The flowchart and the source program listing of the subroutine are given in Fig. VI-20 and Table VII-23, respectively. The subroutine has four arguments. The first argument is a square matrix, and the second and fourth are vectors of order given by the third argument. The square matrix is assumed to be listed columnwise (the number and orders of the vectors being equal to the third argument). The subroutine adds on the vector in the fourth argument the product of the matrix in the first argument by the vector in the second argument. There is no error return in the subroutine.

9. Subroutine ELDI. Subroutine ELDI is called by subroutines S01, S02, and S04. The flowchart and the source program listing of ELDI are given in Fig. VI-21 and Table VII-24, respectively. The subroutine generates in vector PD the direction cosines of the pressure direction for element types 1, 2, and 4. (See description of Input Item 4, Sect. IV-B, Vol. I, and Table III-3, Vol. I, for the rule for determining the pressure direction.) If the element is in the general wind direction, the pressure is set to zero. If an error is encountered, IERR is set to 1, and the subroutine returns control to the calling program.

10. Subroutine PLBE. Subroutine PLBE is called by subroutines S03, and S04. The flowchart and the source program listing of PLBE are given in Fig. VI-22 and Table VII-25, respectively. This subroutine generates, in local coordinates, the free-free stiffness matrix of a grid beam element in $A(6,6)$, which is located in COMMON(200). The matrix may be partitioned with respect to degrees of freedom (nine submatrices, $2 \times 2$ each). In generation of the stiffness matrix, the shear deformations are ignored.

11. Subroutine RLOC. Subroutine RLOC is called by subroutines S02, S03, and S04. The flowchart and the source program listing of RLOC are given in Fig. VI-23 and Table VII-26, respectively. Its function is similar to that of subroutine ADM, described in Sect. III-B. The arguments in this subroutine are all implicit. They are $S$, A, IDS, II, JJ, IR, JR, NY. The subroutine assumes that $S$ is an IDS X IDS matrix; $A$ is a $6 \times 6$ matrix. The objective of the subroutine is to put NY X NY submatrix of matrix $A$ on matrix $S$. The constants II and JJ are the row and column numbers of the first word of NY X NY submatrix of matrix $A$. The constants IR and JR are the row and column numbers of the corresponding word in matrix $S$. In contrast to subroutine ADM, subroutine RLOC does not add, but replaces the entries of matrix $A$ on $S$. After the replacement, the processed portion of $A$ is nullified. There is no error return in the subroutine.

12. Subroutine S01. Subroutine S01 is called by subroutine STFS. The flowchart and the source program listing of S01 are given in Fig. VI-24 and Table VII-27, respectively. The subroutine generates in S the free-free stiffness matrix of element type 1 in the overall coordinate system, and determines constants PRCO, CONS, IPBG, and IPEN for the generation of load vector $P$ (also in overall coordinates) in the main program of Link 2, as described in Sect. III-A. The portion of the load vector related with the temperature change is also handled in the main program of Link 2. To obtain the direction cosines of the unit vector in the pressure direction, subroutine S01 calls subroutine ELDI. When an error condition is encountered, subroutine S01 sets IERR to 1 and returns control to the calling program.

13. Subroutine S02. Subroutine S02 is called by subroutine STFS. The flowchart and the source program listing of S02 are given in Fig. VI-25 and Table VII-28, respectively. The subroutine generates in $S$ the free-free stiffness matrix of element type 2 in the overall coordinate system, determines constants PRCO, CONS, IPBG, IPEN, and modifies vector UV so that load vector $P$ expressed in overall coordinates can be generated by the main program of Link 2 (see Sect. III-A). By calling subroutine CODI, subroutine S02 first generates the direction cosines of the local axes in DIR(3,3). Then subroutine BEAM is called to generate in $A(6,6)$ the free-free stiffness matrix of the element in the local coordinate system. The direction cosines of the direction normal to the element are obtained and stored in PD(3) by means of subroutine ELDI. Then the free-free stiffness matrix, in local coordinates, is carried from $A(6,6)$ into $S$ by means of subroutine RLOC. Finally, by calling subroutine STRA, subroutine S02 obtains and stores in $S$ the description of
the free-free stiffness matrix in the overall coordinate system. The content of COMMON location IERR is transmitted intact to the calling program for error handling.

14. Subroutine S03. Subroutine S03 is called by subroutine STFS. The flowchart and the source program listing of S03 are given in Fig. VI-26 and Table VII-29, respectively. The subroutine generates in S the free-free stiffness matrix of element type 3 in the overall coordinate system, modifies vector UV so that the portion of element load vector P related with the thermal loads can be generated in the main program of Link 2, and generates the remaining portion of element load vector P in the overall coordinate system. By calling subroutine CODI, subroutine S03 generates the direction cosines of the local axes in DIR(3,3). Then, by means of subroutine PLBE, the free-free stiffness matrix of the element in the local coordinate system is obtained and stored in A(6,6). The free-free stiffness matrix in local coordinates is carried from A(6,6) into S by means of subroutine RLOC. By calling subroutine STRA, subroutine S04 obtains and stores in S the description of the free-free stiffness matrix in the overall coordinate system. The load vector due to pressure and acceleration is obtained in the overall coordinate system and stored in P. The distortions of the free-free element due to temperature gradient are first obtained in the local coordinate system, then, by means of subroutine TRAN, in the overall coordinate system, and both descriptions are placed in UVG. Vector UVG is then added to vector UV, so that the main program of Link 2 can handle the thermal portion of P (see Sect. III-A). The content of COMMON location IERR is transmitted intact to the calling program for error handling.

15. Subroutine S04. Subroutine S04 is called by subroutine STFS. The flowchart and the source program listing of S04 are given in Fig. VI-27 and Table VII-30, respectively. The program generates in S the free-free stiffness matrix of element type 4 in the overall coordinate system, and the description of the element load vector P in the overall coordinate system is partly obtained. The remaining portion of element load vector P is obtained in the main program of Link 2. By calling subroutine CODI, subroutine S04 first obtains and stores in DIR(3,3) the direction cosines of the local axes. Then the contributions of the pressure and the acceleration loadings to the description of element load vector P in the overall coordinate system are partly obtained. In obtaining the pressure direction, subroutine S04 calls subroutine ELDI. The stiffness matrix in the local coordinates is obtained in two steps. In the first step, subroutine BEAM is called to obtain the stiffness of the element in the local xy plane for storage in A(6,6). Then this matrix is carried into S by calling subroutine RLOC four times. In the second step, subroutine PLBE is called to obtain the stiffness of the element for deformations out of the local xy plane for storage in A(6,6), and the matrix is carried into S by calling subroutine RLOC once. The description of the free-free stiffness matrix is first obtained in local coordinates, then by means of subroutine STRA, in overall coordinates, for storage in S. So that the thermal load portion of P can be properly obtained in the main program of Link 2, subroutine S04 first computes into UVG the distortions of the free-free element due to temperature gradients, in the local coordinate system. Then, by means of subroutine TRAN, this vector is expressed in the overall coordinate system and added to vector UV. The content of COMMON location IERR is transmitted intact to the calling program for error handling.

16. Subroutine S05. Subroutine S05 is called by subroutines STFS, S11, S13, and S15. The flowchart and the source program listing of S05 are given in Fig. VI-28 and Table VII-31, respectively. When called by STFS (COMMON location IGEM is zero when called by STFS), subroutine S05 generates the free-free stiffness matrix S and partially generates load vector P of element type 5 in the overall coordinate system. When called by subroutines S11 and S13, it generates S and P (partially) in the local coordinates of the element, for element types 11 and 13, respectively, for membrane stretching. Actually, for the latter type of elements, X, Y, Z, XD, YD, and ZD contain local coordinates of the vertices. By calling subroutine TRIM, subroutine S05 first generates [M] and [N] matrices (defined in Ref. 2), in EM and EN locations in COMMON. The [D] matrix (Ref. 2) corresponds to D33 in the subroutine. As far as the free-free stiffness matrix is concerned, the objective of this subroutine is to generate the \([K_w]\) matrix of Eq. (46) in Ref. 2 with \(a = b = c = d = e = 0\). Submatrices \([P]\), \([Q]\), and \([R]\) (Ref. 2) are obtained by executing the triple matrix products by means of subroutine TRM, and then placed into S by means of subroutine ADM. For element types 11 and 13, pressure loading is not considered in this subroutine, but for element type 5, the pressure loading is handled in this routine. In the latter case, if the element is a subelement, the pressure is considered only for the first subelement of both ways of subdivisions (see Sect. III-A). Constants IPBG, IPEN, and CONS for
the handling of the acceleration loading in the main program of Link 2 are generated in this subroutine for all cases. The temperature loading portion of element load vector P is also handled in the main program of Link 2. When an error condition is encountered, the subroutine sets IERR to 1 and returns control to the calling program. The explicit expression of the free-free stiffness matrix may be obtained from Ref. 2.

17. Subroutine S07. Subroutine S07 is called by subroutines STFS and S11. The flowchart and the source program listing of S07 are given in Fig. VI-29 and Table VII-32, respectively. If the calling program is S11, then X, Y, Z, XD, YD, and ZD contain the local coordinates of the vertices; therefore S and P represent the free-free stiffness matrix and the element load vector for bending of element type 11, in local coordinates. If the calling program is STFS, S and P represent the free-free stiffness matrix and the element load vector of element type 7, in the overall coordinate system. The portion of P related with pressure loading is generated by subroutine S07. The constants IPBG, IPEN, and CONS are generated by subroutine S07 so that the portion of the vector P related with acceleration loading can be handled in the main program of Link 2 (see Sect. III-A). The portion of P related with the thermal loads is also handled in the main program of Link 2. Subroutine S07 generates in vector UV the distortions of the free-free element due to temperature gradient (see Sect. III-A). By calling subroutine TRM, subroutine S07 first obtains \[M\], \[N\], and \[L\] matrices of Ref. 2 in locations EM, EN, and EL. The matrices \[D\] and \[D'\] of this reference correspond to D33 and E22 in the subroutine. The triple matrix products indicated by Eqs. (45) and (51) of Ref. 2 are carried out by means of subroutine TRM, and are properly placed into S by means of subroutine ADM. The objective of subroutine S07 in generating S is to obtain the shaded portions of \([K_a]\) and \([K_b]\) matrices given by Eqs. (49) and (55) of Ref. 2. In generating the shaded portion of \([K_b]\) given by Eq. (55) of Ref. 2, the subroutine uses the “constant trace scheme” of Ref. 3. When an error condition is detected, the subroutine sets IERR to 1 and returns control to the calling program.

18. Subroutine S09. Subroutine S09 is called by subroutine STFS. The flowchart and the source program listing of S09 are given in Fig. VI-30 and Table VII-33, respectively. The objective of this subroutine is to compute the free-free stiffness matrix and the element load vector of element type 9, in the overall coordinate system, into locations S and P. The portion of P related with压力 loading is generated in subroutine S09. The subroutine generates the values of IPBG, IPEN, and CONS values for the handling of the acceleration loading portion of P in the main program of Link 2, which also handles the thermal load portion. The submatrices of the free-free stiffness matrix are obtained in the form of triple matrix products computed by means of subroutine TRM. These submatrices are properly placed in S by means of subroutine ADM. If the volume of the element is too small relative to a reference volume, an error message is printed out and the generation of S and P is skipped. If an error condition is encountered during the execution of the subroutine, IERR is set to 1, and control is returned to the calling program.

19. Subroutine S11. Subroutine S11 is called by subroutine STFS. The flowchart and the source program listing of S11 are given in Fig. VI-31 and Table VII-34, respectively. The objective of this subroutine is to generate the free-free stiffness matrix and partially generate the element load vector in S and P, respectively, of element type 11, in the overall coordinate system. The matrix in S is that of Eq. (61) of Ref. 2, with \[a = b = c = d = e = 0\]. Subroutine S11 first calls subroutine CORT to obtain and store in X, Y, Z, XD, YD, ZD, the coordinates of the vertices in the local coordinate system, and the direction cosines of the local axes in DIR(3,3). Next, by calling subroutine S07, subroutine S11 generates the bending portion of S and P by assuming the order as 9. Next, the quantities in S and P are properly relocated so that S and P are of order 18. The same relocation is applied to vector UV, which is generated by subroutine S07. After this, subroutine S11 calls subroutine S05 to generate the membrane portion of S and P. The P vector is partially generated in subroutine S11. The acceleration load and the thermal load portions of P are handled in the main program of Link 2, which also handles the thermal load portion. The submatrices of the free-free stiffness matrix are obtained in the form of triple matrix products computed by means of subroutine TRM. These submatrices are properly placed in S by means of subroutine ADM. If the volume of the element is too small relative to a reference volume, an error message is printed out and the generation of S and P is skipped. If an error condition is encountered during the execution of the subroutine, IERR is set to 1, and control is returned to the calling program.

20. Subroutine S13. Subroutine S13 is called by subroutine STFS. The flowchart and the source program listing of S13 are given in Fig. VI-32 and Table VII-35, respectively. The objective of this subroutine is to generate the free-free stiffness matrix and partially generate the element load vector in S and P, respectively, of element type 13, in the overall coordinate system. Subroutine S13 first calls subroutine CORT to obtain and store...
in X, Y, Z, XD, YD, ZD the coordinates of the vertices in the local coordinate system, and the direction cosines of the local axes in DIR(3,3). Next, subroutine S13 calls subroutine S05 to generate the membrane rigidity and the corresponding load vector in S and P. The pressure load portion of P is generated in subroutine S13, and the acceleration loading portion and the thermal load portion of P are generated in the main program of Link 2 (see Sect. III-A). Having generated S and P in local coordinates, subroutine S13 calls subroutine TRAN to express P in the overall coordinate system, and subroutine STRA to express S in the overall coordinate system. The content of COMMON location IERR is transmitted intact to the calling program for error handling.

21. Subroutine S15. Subroutine S15 is called by subroutine STFS. The flowchart and the source program listing of S15 are given in Fig. VI-33 and Table VII-38, respectively. The objective of this subroutine is to generate the free-free stiffness matrix and the element load vector of element type 15, in locations S and P, expressed in the overall coordinate system. Only the thermal load portion of P is generated in the main program of Link 2 (see Sect. III-A). The generation of the free-free stiffness matrix is performed as described in Ref. 4. The terminology of Table VII-36 should be interpreted in the light of Ref. 5. Before returning control to the calling program, subroutine S15 calls subroutine TRM to obtain the EM and EN arrays corresponding to M and N, respectively, of Ref. 4. The triple matrix products of Ref. 4 are performed by means of subroutine TRM, and properly placed in S by means of subroutine ADM. The first term in Eq. (9) of Ref. 4 is obtained in S by means of subroutine S05. In case of error, IERR location is set to 1 and control is returned to the calling program.

22. Subroutine S17. Subroutine S17 is called by subroutines STFS and S18. The flowchart and the source program listing of S17 are given in Fig. VI-34 and Table VII-37, respectively. The objective of this subroutine is to generate the free-free stiffness matrix and the element load vector of element type 17, in locations S and P, expressed in the overall coordinate system. Only the thermal load portion of P is generated in the main program of Link 2 (see Sect. III-A). The generation of the free-free stiffness matrix is performed as described in Ref. 5. The triple matrix products of this reference are performed by means of subroutine TRM, and properly placed in S by means of subroutine ADM. The terminology in Table VII-37 should be interpreted in the light of Ref. 5. In case of error, location IERR is set to 1 and control is returned to the calling program.

23. Subroutine S18. Subroutine S18 is called by subroutine STFS. The flowchart and the source program listing of S18 are given in Fig. VI-35 and Table VII-38, respectively. The objective of this subroutine is to generate the free-free stiffness matrix and the element load vector of element type 18, in locations S and P, expressed in the overall coordinate system. Only the thermal load portion of P is generated in the main program of Link 2 (see Sect. III-A). The generation of the free-free stiffness matrix is performed as described in Ref. 5. For this purpose, subroutine S18 first calls subroutine S17 to generate the membrane portion of S and P. The triple matrix products of Ref. 5 are performed by means of subroutine TRM, and properly placed in S by means of subroutine ADM. The terminology in Table VII-38 should be interpreted in the light of Ref. 5. Before returning control to the calling program, subroutine S18 modifies vector UV for the inclusion of thermal gradient effects. The content of COMMON location IERR is transmitted intact to the calling program for error handling.

24. Subroutine STFS. Subroutine STFS is called by the main program of Link 2. The flowchart and the source program listing of STFS are given in Fig. VI-36 and Table VII-39, respectively. The subroutine has one argument, which is the type number of the current element being processed by the calling program. The function of this subroutine is to call the proper subroutine from among S01, S02, S03, S04, S05, S07, S09, S11, S13, S15, S17, and S18 to suit the type number in the argument. The functions of these subroutines are to generate the free-free stiffness matrix, and partially generate the element load vector, in locations S and P, expressed in the overall coordinate system. There is no error return in the subroutine.

25. Subroutine STRA. Subroutine STRA is called by subroutines S02, S03, S04, S11, and S13. The flowchart and the source program listing of STRA are given in Fig. VI-37 and Table VII-40, respectively. The objective of this subroutine is to generate in S the description of the free-free stiffness matrix, in overall coordinates, from the description in local coordinates in S, and the direction cosines of local axes in DIR(3,3). The subroutine assumes that S is an IDS × IDS matrix. By calling subroutine TRAN, IDS times, subroutine STRA first obtains and places the description of each of the IDS vectors of S, in overall coordinates, in the same S locations. Then it generates in S the transpose of IDS × IDS free-free stiffness matrix. Finally, by calling subroutine TRAN, again IDS times, subroutine STRS obtains and
places the description of each of the IDS vectors of the transposed matrix, in overall coordinates, in the same S locations. The final matrix is the description of the free-free stiffness matrix in the overall coordinate system. There is no error return in the subroutine, and all arguments are implicit.

26. Subroutine TICK. Subroutine TICK is called by the main program of Link 2. It is identical with subroutine TICK of Link 1. For further information, see Sect. II-B-13. The source program listing of this program is given in Table VII-41.

27. Subroutine TOPO. Subroutine TOPO is called by the main program of Link 2. The flowchart and the source program listing of TOPO are given in Fig. VI-38 and Table VII-42, respectively. The objective of this subroutine is to extract and analyze the descriptive words of the element being currently processed by the main program of Link 2. The subroutine is identical with subroutine TOPO of Link 1 up to the statement whose number is 1450 (see Fig. VI-38). As a result of the analysis of the descriptive words, the vertex labels and the property type numbers of the element are obtained in N block and in locations IELT, IMET, JPRS, ITIC, ITEM, JS1Y, JSDZ, JMMX, JMMY, JMMZ, JMF1, JARE, respectively. In case of error, the subroutine returns control to the calling program.

28. Subroutine TRAN. Subroutine TRAN is called by subroutines S03, S04, S11, S13, and STRA. The flowchart and the source program listing of TRAN are given in Fig. VI-39 and Table VII-43, respectively. The subroutine has two explicit arguments. The objective of this subroutine is to generate the description of a vector of order (IGEM + 1)*IMS*3 in the overall coordinates from the description of the vector in the local coordinates, and DIR(3,3) (the directions cosines of local axes). The description of the vector in the local coordinate system is in the array indicated by the first argument, just after the entry indicated by the second argument. The subroutine first computes the description of the vector in the overall system in DUM block, and then carries it on the local description. There is no error return in the subroutine.

29. Subroutine TRIM. Subroutine TRIM is called by subroutines S05, S07, and S15. The flowchart and the source program listing of TRIM are given in Fig. VI-40 and Table VII-44, respectively. The objective of this subroutine is to obtain in blocks EM, EN, and EQ the matrices [M], [N], and [L] of Ref. 2 from the information in XD and YD. There is no error return in the subroutine.

30. Subroutine TRM. Subroutine TRM is called by subroutines S05, S07, S09, S15, S17, and S18. The flowchart and the source program listing of TRM are given in Fig. VI-41 and Table VII-45, respectively. The objective of the subroutine is to perform triple matrix products of the type \([B]^T[A][B]\) or \([C^T][A][B]\) where \([A]\) is always a symmetric matrix of order 3 or less, and \([B]\) and \([C]\) matrices of order \((3\times 4)\) or less. The subroutine has five arguments. If the last argument is negative, \([C^T][A][B]\) is performed; if the last argument is positive, \([B]^T[A][B]\) is performed. The order of the symmetric matrix \([A]\) is given by the fourth argument. The absolute value of the last argument is the column order of \([C]\) or \([B]\). The matrices \([A]\), \([B]\), and \([C]\) are indicated by the first, second, and third arguments, respectively. The subroutine returns control to the calling program by placing the triple product into the array indicated by the third argument. There is no error return in the subroutine.
A. Main Program of Link 3

The flowchart and the source program listing of the main program of Link 3 are given in Fig. VI-42 and Table VII-46, respectively. The logical function of the program may be summarized as follows:

1. The program generates and stores the upper decomposed stiffness matrix in the IST-pointer-related vector, and the unknown deflections in the IDEF-pointer-related vector.

2. Possibly destroying some portions of the decomposed stiffness matrix, the program generates in BB array the complete list of nodal deflections, and carries them onto the IDEF-pointer-related vector.

3. If execution of the stress link is requested, i.e., if INX = 4, the program computes into the IST-pointer-related vector the forces acting on mesh points (see Output Item 20, Sect. VI-D, Vol. I).

4. If INX = 4, the program generates in the IST-pointer-related vector the list of labels of the elements meeting at the mesh points, immediately after the residual forces computed in (3), and saves this list on tape ITAS for use in Link 4.

5. Depending upon the values of INX and ITAS, the main program transfers the control either to Link 4 or to Link 1, as the logically last operation.

In carrying out function (1), the program calls subroutine VELAS, which requires as arguments the number of equations in the system, the pointer of the list of pointers of the diagonal elements of the coefficient matrix, the pointer of the coefficient matrix, and the pointer of the right-hand-side vector. The successful solution of linear equations is indicated by the zero content of the second argument. Function (2) is carried out with the help of the information in IBO-, IBB- and IIC-pointer-related arrays and within the framework of Table III-1, Vol. I. The program produces Output Item 19 from BB block, and calls subroutine PUNC for other modes of output (see Sect. V-F, Vol. I). Then, the information in BB block is carried out to the IDEF-pointer-related vector for use in Link 4. To carry out function (3), the program calls subroutine RESI, and to produce Output Item 20, it calls subroutine RESW. Function (4) is carried out by means of subroutine ELST. The main program, in measuring the elapsed time in executing Link 3, and solving the linear equations, calls subroutine TICK. Output Items 18, 19, and 21 are directly produced by the main program.
B. Subroutines of Link 3

1. Subroutine ELST. Subroutine ELST is called by the main program of Link 3. The flowchart and the source program listing of ELST are given in Fig. VI-43 and Table VII-47, respectively. The function of the subroutine is to generate, for each node, information listing the labels of the non-one-dimensional elements meeting at a node. This information is listed as a one-dimensional array starting immediately after the residual forces produced by subroutine RESI (between points E and E' in Fig. III-1, Vol. II, basic). For this purpose, 13 words are assigned for every mesh point. The first word contains the number of non-one-dimensional elements meeting at the mesh point, and the remaining words the labels of these elements. Whenever there are more than 12 non-one-dimensional elements meeting at the mesh point, subroutine ELST returns control to the calling program by setting ITAS to zero, thus preventing the execution of Link 4 even if INX = 4. When the number of non-one-dimensional elements meeting at a mesh point and their labels are obtained successfully, the subroutine generates one logical record on tape ITAS for each mesh point to contain such information, and thus releases the corresponding core area. These records are listed after the elemental matrices, and are ordered with the labels of the mesh points. The subroutine also counts the one-dimensional elements and saves the total in COMMON IELT, which is generated by the main program of Link 3. The standard ELAS computation with increasing mesh-point labels and conforms with Table III-3, Vol. I (i.e., the residual forces of the first mesh point are to be listed first, the residual forces of the second mesh point are to be listed second, etc.). Since tape ITAS is already positioned by the main program of Link 2 for this purpose, the subroutine reads in sequentially the element matrices (the stiffness matrix and the load vector without thermal load contribution) one at a time, and performs the operation of “element load vector less element stiffness matrix times vertex deflections,” and assembles the resulting vector onto the vector of residual forces. In the case of subelements, the scaling factors discussed in Section III-A are properly considered. If an error is detected during the tape handling, the subroutine sets the contents of ITAS to zero and returns control to the calling program, thus preventing the execution of Link 4 even if INX = 4. The residual forces are kept intact in the core until the execution of Link 4 is completed.

2. Subroutine PUNC. Subroutine PUNC is called by the main program of Link 3. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-48. If the user wishes to produce Output Item 19 in different media and format, he may do so by writing his version of this subroutine, as explained in Sect. V-F, Vol. I. The logical function of this subroutine is to copy deflections from BB block to the desired output media with the desired format.

3. Subroutine RESI. Subroutine RESI is called by the main program of Link 3. The flowchart and the source program listing of RESI are given in Fig. VI-44 and Table VII-49, respectively. The function of this subroutine is to generate the forces acting at the mesh points, in the overall coordinate system, in the IST-pointer-related vector. Such forces consist of nonthermal element forces less the elastic forces (element stiffness matrix times the vertex deflections). In the absence of thermal loading, these forces represent the round-off errors at a nonboundary point, and the reaction force at a boundary point where the deflections are prescribed, or prescribed concentrated loads where the deflections are not prescribed. In this context, these forces are labelled as the residual forces. The residual forces are used in Link 4 to compute average stresses at the boundary points. To compute the residual forces, subroutine RESI clears the first IND words of the IST-pointer-related vector, and considers that the residual forces are to be listed as in Table III-3 with increasing mesh-point labels and conforms with Table III-1, Vol. I (i.e., the residual forces of the first mesh point are to be listed first, the residual forces of the second mesh point are to be listed second, etc.). Since tape ITAS is already positioned by the main program of Link 2 for this purpose, the subroutine reads in sequentially the element matrices (the stiffness matrix and the load vector without thermal load contribution) one at a time, and performs the operation of “element load vector less element stiffness matrix times vertex deflections,” and assembles the resulting vector onto the vector of residual forces. In the case of subelements, the scaling factors discussed in Section III-A are properly considered. If an error is detected during the tape handling, the subroutine sets the contents of ITAS to zero and returns control to the calling program, thus preventing the execution of Link 4 even if INX = 4. The residual forces are kept intact in the core until the execution of Link 4 is completed.

4. Subroutine RESW. Subroutine RESW is called by the main program of Link 3, if INP ≠ 0. The flowchart and the source program listing of RESW are given in Fig. VI-45 and Table VII-50, respectively. The purpose of this subroutine is to produce Output Item 20. This is achieved by looping on mesh points. At every loop cycle, the subroutine first abstracts the residual force of the respective mesh point from the list of residual forces in the IST-pointer-related vector, and arranges the components to a complete six-component vector in accordance with Table III-1, Vol. I, and finally prints a line for these components. The ordering of the residual forces is explained in Table III-3, Vol. II (basic). In arranging the components of a mesh point, the subroutine uses the constant IELT, which is generated by the main program of Link 3 for Output Item 19 (the ordering of Output Items 19 and 20 is similar). There is no error return in the subroutine.
5. Subroutine TICK. Subroutine TICK is called by the main program of Link 3. It is identical with subroutine TICK of Link 1. For further information see Section II-B-13. The source program listing of this program is given in Table VII-51.

6. Subroutine VELAS. Subroutine VELAS is called by the main program of Link 3. The flowchart and the source program listing of VELAS are given in Fig. VI-46 and Table VII-52, respectively. The purpose of this subroutine is to solve linear equations with positive-definite, symmetric, and variable-banded coefficient matrices. The subroutine has four explicit and no implicit arguments. The first argument is the order of the linear system (i.e., the number of equations); at entry to the program, the second argument contains the pointer of the vector listing the pointers of the diagonal elements of the coefficient matrix; the third argument is the pointer of the coefficient matrix; the fourth argument is the pointer of the right-hand-side vector. Subroutine VELAS assumes that the arrays related with the last three arguments are all in COMMON. By applying the Cholesky scheme, the subroutine first obtains the decomposed matrix (referred to in Fig. VI-46 as U(I,J)) on the coefficient matrix (referred to in Fig. VI-46 as A(I,J)); then by a forward sweep it obtains the auxiliary solution (referred to in Fig. VI-46 as Y(I)) in the right-hand-side vector (referred to in Fig. VI-46 as B(I)); and finally, by a backward sweep, it obtains the solution vector (referred to in Fig. VI-46 as X(I)) in the right-hand-side vector (referred to in Fig. VI-46 as B(I)). During decomposition, if the quantity under the radical sign is nonpositive, the subroutine returns control to the calling program by setting the location of the associated diagonal element relative to the beginning of the coefficient matrix in the second argument. If the first argument is nonpositive, the return is made by setting the second argument to -1. The subroutine assumes that a quantity is positive if it is larger than the $10^{-16}$ multiple of the smallest diagonal element (in magnitude) of the coefficient matrix. During decomposition, the subroutine uses NN locations immediately after the array related with the third argument to store the number of matrix elements in the columns of coefficient matrix within the shaded area shown in Fig. II-1, Vol. I. Here NN is the order of the system. The subroutine assumes that the coefficient matrix is arranged as described in Table III-3 under IST-pointer-related vector.
V. Main Program and Subroutines of Link 4

A. Main Program of Link 4

The flowchart and the source program listing of the main program of Link 4 are given in Fig. VI-47 and Table VII-53, respectively. The logical functions of the program may be summarized as follows:

1. For line elements, by means of subroutine DIMI, the program computes stress resultants at the end points of the elements and prints out Output Item 24.

2. For non-one-dimensional elements, it computes the stresses at the mesh points and prints out Output Item 22, by means of Link 4 programs other than DIMI.

3. It transfers control to the main program of Link 1 as the logically last operation.

By checking the contents of IONE (see Sect. IV-B-1), the main program performs either function (1) or (2) or both.

If function (2) is to be performed, the program checks whether the starting point of vector FF (see Tables III-4 and III-5, Vol. II, basic) and point E (see Fig. III-1, Vol. II, basic) are overlapping. If overlapping occurs, Error Message 23 is produced and no stress computation is done. Otherwise the main program loops on mesh points with the objective of computing stresses at a mesh point for each material group and for each class group (see Output Item 22 in Sect. VI-E, Vol. I). When the loop on mesh points is satisfied, if there are line elements in the structure, the main program calls subroutine DIMI for function (1). During the performance of function (2), the program calls subroutines ABEQ, BOFI, CAS4, DINA, FINDQ, FINDX, GENE, INLZ, LEST, MDIN, META, SAME, SETA, and STRS. The program calls subroutine TICK to measure the time spent in output Output Item 22, by means of Link 4 programs Link 4. The main program is directly responsible for the production of Output Items 22 and 25.

B. Subroutines of Link 4

1. Subroutine ABEQ. Subroutine ABEQ is called by the main program of Link 4, if the current mesh point ICN is on the boundary of a two- or three-dimensional continuum. The flowchart and the source program listing of ABEQ are given in Fig. VI-48 and Table VII-54, respectively. The objective of this subroutine is to generate
the first IEQ rows of the augmented matrix A (see Table III-5, Vol. II, basic), the corresponding weights in vector IWG, and the actual values of the prescribed stresses in vector SR. The first IEQ rows of the augmented matrix A correspond to the IEQ number of stress boundary conditions at the boundary point ICN. These equations are generated as discussed in Sect. II, Vol. II (basic). The subroutine first copies the residual vector (see Sect. IV-B-3) of mesh point ICN into RES vector. Then, depending upon the class type of current ICth group, it computes certain parameters listed in Table V-1. The program carries two right-hand-side vectors for class 6 and 8 structures in the strain deflection equations, because of the symmetry in strains and curvature changes (see Ref. 2). This is very useful in minimizing the column order of the strain deflection equations.

In Table V-1, the parameters used by the subroutine are defined. The meanings of these parameters are as follows: IEQ is the number of the stress boundary equations (note that the number of stress boundary conditions is the product of IEQ*IRIG); IRIG is the number of right-hand sides in the strain deflection equations; ICOL is the column order of the coefficient matrix of strain deflection equations; vector N lists the component number (see Table VI-6, Vol. I) of the prescribed stress (as stated in Sect. VI-E, Vol. I, the local coordinate axes on a boundary point are such that the first axis is the outer normal of the boundary surface, and the second and the third axes are tangential to the boundary surface; the stress boundary conditions are expressed in the local coordinate system); IREB and IREN are the entry numbers of the beginning and the end, respectively, of the portion of vector RES to be used in generating the prescribed stress values for the right-hand sides of the stress boundary condition equations. Because of the ordering of the residual forces (first, force components, and then moment components) for the second right-hand side, the program takes IREB and IREN as IREB + 1 and IDEG, respectively. The first and second columns of matrix NEK list the labels of the local axes to be used in projecting the portion of vector RES for obtaining the prescribed stress components, for the first and second right-hand sides, respectively. Matrix REK, like matrix NEK, indicates whether any sign change is to be performed for the correct sign of the prescribed stress. Scale factors CR and CL are used in scaling a stress boundary condition equation to achieve similar orders of magnitude in the whole set strain deflection equations; CR is for the right-hand side, CL for the left-hand side. The basic format of a stress boundary condition equation is shown in Table V-1. To obtain the components of the best-fit strain tensor from the strain deflection equations in a manner that satisfies the stress boundary conditions more correctly than the remaining equations, the subroutine assigns the stress boundary equations a weight of 100 in the corresponding entries of vector IWG. The subroutine also updates the equation count ICON, and saves the prescribed stress values in vector SR in the order shown in Table V-2. There is no error return in the subroutine.

2. Subroutine ACEL. Subroutine ACEL is called by subroutine DINA if the IPIR field of the control card (see Table IV-2, Vol. I) is 2. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-55. If the user wishes to prescribe local coordinate systems at the mesh points of shell structures, he may do so by writing his version of this subroutine, as explained in Sect. V-E, Vol. I. The logical function of this subroutine is to define matrix DIN for the direction cosines of the local axes of mesh point ICN.

3. Subroutine BEST. Subroutine BEST is called by subroutines BOFI and QUAD. The flowchart and the source program listing of BEST are given in Fig. VI-49 and Table VII-56, respectively. The objective of this subroutine is to obtain the direction cosines of the normal of the best-fit plane (in the least squares sense) related with the mesh points listed in the array referenced by the second argument and mesh point ICN. The number of mesh points listed in the second argument is given by the third argument. The subroutine places the direction cosines of the normal into the array indicated by the first argument. One condition equation is generated for each mesh point listed in the array referenced by the second argument to express the situation for that point to be in the sought-for plane. The mesh-point coordinates are obtained by means of subroutine FINDX. The equation of the plane is arbitrarily expressed in a coordinate system that is parallel to the overall, but located at a point with coordinates 1.15, 1.16, and 1.17 less than those of mesh point ICN. Once the condition equations are established, the coefficients of variables (i.e., quantities proportional to the direction cosines of the normal) are solved by least squares, by first premultiplying both sides of the condition equations by the transpose of the coefficient matrix, and then solving the resulting equations by means of subroutine INV. If the inversion fails in subroutine INV, subroutine BEST attempts to approximate the direction cosines of the normal, as explained in block 45 of Fig. VI-49. This latter process necessitates a vector product, which is carried out by means of subroutine VECT.
Table V-1. Values of important parameters used in subroutine ABEQ for various classes (see Sect. V-B-1 for discussion)

<table>
<thead>
<tr>
<th>Class No.</th>
<th>IEQ</th>
<th>N(1)</th>
<th>N(2)</th>
<th>N(3)</th>
<th>IREB</th>
<th>IREN</th>
<th>IRIG, (NES(2))</th>
<th>NEK(I,1)</th>
<th>NEK(I,2)</th>
<th>REK(I,1)</th>
<th>REK(I,2)</th>
<th>ICOL, (NES(1))</th>
<th>CR</th>
<th>CL</th>
<th>IDR, (NES(3))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>[1]</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>3</td>
<td>ARE</td>
<td>ARE³</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>3</td>
<td>-TE*ARE</td>
<td>ARE²</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>3</td>
<td>XE*DD,1</td>
<td>ARE²</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>1</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>6</td>
<td>DD,1</td>
<td>DD,1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>[1]</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>1</td>
<td>ARE</td>
<td>ARE³</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>[2]</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>1</td>
<td>XE*DD,1</td>
<td>DD,1</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>3</td>
<td>ARE</td>
<td>ARE²</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>[2]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>[1]</td>
<td>3</td>
<td>ARE</td>
<td>ARE³</td>
</tr>
</tbody>
</table>

Table V-2. Arrangement of prescribed boundary forces by subroutine ABEQ in SR vector for the eight class types

<table>
<thead>
<tr>
<th>Class</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SR(1)</td>
<td>σₐ</td>
<td>Mₐ</td>
<td>σₐ</td>
<td>Mₐ</td>
<td>Nₐ</td>
<td>Nₐ</td>
<td>Nₐ</td>
<td>Nₐ</td>
</tr>
<tr>
<td>SR(2)</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>Nₑθ</td>
<td>Nₑθ</td>
</tr>
<tr>
<td>SR(3)</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>Nₑθ</td>
<td>Nₑθ</td>
</tr>
<tr>
<td>SR(4)</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>τₑθ</td>
<td>Mₑθ</td>
<td>Nₑθ</td>
<td>Nₑθ</td>
</tr>
</tbody>
</table>

*See Table VI-6, Vol. I.
Before returning control to the calling program, subroutine BEST calls subroutine UNIT to normalize the vector in the first argument. There is no error return in the subroutine.

4. Subroutine BOFI. Subroutine BOFI is called by the main program of Link 4. The flowchart and the source program of BOFI are given in Fig. VI-50 and Table VII-57, respectively. The objective of this subroutine is to determine from the element set information in NEL whether mesh point ICN is on the boundary. If mesh point ICN is not on the boundary, the subroutine returns control to the calling program without any action. However, if mesh point ICN is found to be on the boundary, the subroutine sets INBON = 1 and AST = 1H*, and computes the direction cosines of the outer normal of the boundary at mesh point ICN into vector BIR. To obtain the direction cosines, subroutine BOFI calls subroutine INER to obtain a general vector heading towards the structure, and calls subroutine BEST to obtain the direction cosines of the best-fit plane to boundary nodes neighboring mesh point ICN. After redirecting the normal of the plane with the general vector heading towards the structure, the normal of the best-fit plane is assumed to be the outer normal of the structure at the boundary node. If any trouble arises in finding the outer normal, subroutine BOFI will assume that mesh point ICN is an internal one. The subroutine also generates in ARE an average boundary surface area if mesh point ICN is on the boundary. The number of repeated interelement boundaries in the element set and the number of unrepeated element boundaries are used in determining whether mesh point ICN is on the boundary. In performing its function, BOFI also calls subroutines FINDX, SCAL, and UNIT. There is no error return in the subroutine.

5. Subroutine CAS4. Subroutine CAS4 is called by the main program of Link 4. The standard ELAS contains only the dummy version of this subroutine, as shown in Table VII-58. If Output Item 23 is to be produced selectively, this subroutine should be rewritten by the user, as explained in Sect. V-II, Vol. I. The logical function of this subroutine is to change the value of INP as desired in the DO loop on mesh points in the main program of Link 4.

6. Subroutine CODI. Subroutine CODI is called by subroutine DIMI. It is identical with subroutine CODI of Link 2. For further information, see Section III-B-4. The source program listing of this subroutine is given in Table VII-59.

7. Subroutine DIMI. Subroutine DIMI is called by the main program of Link 4, if IONE is positive. The flowchart and the source program listing of DIMI are given in Fig. VI-51 and Table VII-60, respectively. The objective of this subroutine is to obtain the end forces of the one-dimensional elements in the local coordinate system and to produce Output Item 24. For this purpose, the subroutine rewinds tape ITAS and processes the mesh elements one at a time. If a one-dimensional element is encountered, by means of subroutines CODI, STRA and TRAN, the element stiffness matrix, the element load vector, and the deflection vector of the vertices are first expressed in local coordinates of the element (see Fig. VI-1, Vol. I), and then the end forces are expressed as the product of the element stiffness matrix and the vector of vertex deflections less the element load vector. If an error is encountered during the tape handling, the subroutine sets IERR = 1, ICN = the element number, and the explicit argument of the subroutine with the record number as read from the tape, and returns control to the calling program.

8. Subroutine DINA. Subroutine DINA is called by the main program of Link 4, if the current class group is of shell type (i.e., ICAS is equal to or larger than 5). The flowchart and the source program listing of DINA are given in Fig. VI-52 and Table VII-61, respectively. The objective of this subroutine is to determine the direction cosines of the axes in which the stresses are to be expressed in DIN and find the value of ANGLE associated with these axes. If the IPIR value of the control card (see Table IV-2, Vol. I) is larger than 1, subroutine DINA calls subroutine AGEL to do the job. Otherwise, subroutine DINA calls either subroutine QUAD or subroutine REVO, depending upon whether the shell is general or axisymmetrical type, respectively, to perform its function. There is no error return in the subroutine.

9. Subroutine EPAN. Subroutine EPAN is called by subroutine QUAD, if there are less than eight neighboring mesh points in the element set of mesh point ICN (subroutine QUAD is called only for the general shell case). The flowchart and the source program listing of EPAN are given in Fig. VI-53 and Table VII-62, respectively. The objective of this subroutine is to expand the vector containing the labels of the mesh points in the immediate neighborhood of the current mesh point ICN by the element set information of the mesh points in the immediate neighborhood of mesh point ICN. When
control is transferred to the subroutine, vector NSET contains the labels of the immediate neighbors of mesh point ICN, and NB contains the order of this vector. By minimum tape handling, the subroutine obtains the element set information of each of these immediate neighbors from tape ITAS and expands vector NSET with the mesh-point labels of the neighbors of neighbors of mesh point ICN and updates NB correspondingly. Before it returns control to the calling program, the subroutine repositions tape ITAS to the position at the time of entry. There is no error return in the subroutine.

10. Subroutine FINDQ. Subroutine FINDQ is called by the main program of Link 4 and by subroutine SETA. The flowchart and the source program listing of FINDQ are given in Fig. VI-54 and Table VII-63, respectively. The function of this subroutine is to compute, in overall coordinates, the deflection components of the mesh point indicated by the first argument into the vector indicated by the second argument. There is no error return in this subroutine.

11. Subroutine FINDX. Subroutine FINDX is called by the main program of Link 4 and subroutines BEST, BOFI, INER, INLZ, QUAD, REVO, and SETA. The flowchart and the source program listing of FINDX are given in Fig. VI-55 and Table VII-64, respectively. The function of this subroutine is to compute the overall coordinates of the mesh point indicated by the first argument into the vector indicated by the second argument. There is no error return in this subroutine.

12. Subroutine GENE. Subroutine GENE is called by the main program of Link 4. The flowchart and the source program listing of GENE are given in Fig. VI-56 and Table VII-65, respectively. The objective of this subroutine is to generate the columns of matrix NEL, other than the first (the first column of NEL is already generated by the main program of Link 4), the corresponding matrix MAC, vector ICLAS, and variable IMEL. The format of NEL and MAC matrices is given in Table VI-7, Vol. I. Since the rows of matrix NEL contain information about the elements meeting at mesh point ICN, subroutine GENE calls subroutine TOPO to extract such information from COMMON. Matrix MAC is generated by the information in completed matrix NEL. Vector ICLAS and variable IMEL are generated as a by-product of the generation of matrix MAC. Error conditions that may be encountered during the generation of matrix MAC may cause the production of Error Messages 18, 19, and 20. There is no error return in the subroutine.

13. Subroutine INER. Subroutine INER is called by subroutine BOFI, if mesh point ICN is on the boundary. The flowchart and the source program listing of INER are given in Fig. VI-57 and Table VII-66, respectively. The objective of this subroutine is to generate the components of a vector heading towards the structure at mesh point ICN, and store them in the vector indicated by the argument. At the time of entry to the subroutine, vector NSET contains the labels of the neighboring mesh point. The subroutine simply adds the vectors, joining mesh point ICN to its neighbors to obtain the required vector. The mesh-point coordinates are obtained by means of subroutine FINDX. There is no error return in the subroutine.

14. Subroutine INLZ. Subroutine INLZ is called by the main program of Link 4. The flowchart and the source program listing of INLZ are given in Fig. VI-58 and Table VII-67, respectively. The function of this subroutine is to initialize the values of IROT, BST, DIN, W, TE, DT, DG, ICOL, IRIG, IDR, ANGLE, ICON, IERR, and BAS quantities for the stress computation corresponding to the current values of ICN/IM/IC (see Table III-4). In initializing these values, the subroutine assumes that the mesh point is an internal one and the local coordinate system is parallel to the overall coordinate system. Values of TE, DT, and DC are obtained as the arithmetical average of those of the related mesh elements. Values of IRIG, ICOL, and IDR are obtained, depending upon the class type number of the current mesh-element group (see Tables III-4, Vol. II, basic, and V-1). To perform its functions, INLZ calls subroutines FINDX and UNIT. There is no error return in the subroutine.

15. Subroutine INV. Subroutine INV is called by subroutines BEST, LEST, QUAD, and REVO of Link 4. The flowchart and the source program listing of INV are given in Fig. VI-59 and Table VII-68, respectively. The purpose of the subroutine is to solve a set of linear equations by Gauss elimination. The coefficient matrix is referred by the first argument, and the right-hand-side vectors are referred by the third argument. The second argument is the order of the system, and the fourth argument is the number of right-hand-side vectors. When the subroutine returns control to the calling program, the fifth argument contains the value of the determinant of the coefficient matrix, the first argument contains the inverse of the coefficient matrix, and the third argument contains the solution vector if the fourth argument and the determinant are nonzero. This subroutine is borrowed from
the IBM 1620 library of the Jet Propulsion Laboratory as of September 1966.

16. Subroutine LEST. Subroutine LEST is called by the main program of Link 4. The flowchart and the source program listing of LEST are given in Fig. VI-60 and Table VII-69, respectively. The objective of this subroutine is to obtain the components of the best-fit strains from the strain deflection equations. At the time of entry to the subroutine, matrix A contains the augmented matrix of the strain deflection equations, ICON contains the number of equations, JMM the number of columns in the coefficient matrix, JMR the number of right-hand sides (therefore JMX = JMM + JMR is the column order of the augmented matrix), and IWG the weight assigned to each of the strain deflection equations. Considering the multiplicity of the equations as given in vector IWG, the subroutine premultiplies both sides of the strain deflection equations by the transpose of the coefficient matrix, then calls subroutine INV to obtain the best-fit strain components, and reorders the components in matrix C and redefines JMM and JMR such that the first column contains the usual strain components and the second column contains the angular strain (curvature change) components. For axisymmetrical structures, cases in which mesh point ICN is on the axis of revolution are handled separately in the subroutine by considering the thermal strains, if there are any. For shells of revolution, if mesh point ICN is on the axis, vectors DDIS, DROT, and DCOR contain the relative displacement, relative rotation, and relative coordinate of the opposite end of the nodal line with respect to mesh point ICN. These vectors are expressed in the local coordinate system of mesh point ICN. If the inversion performed by subroutine INV is not successful (indicated by zero determinant), subroutine LEST sets IERR = 1 and returns control to the calling program. If INP is 2, the best-fit strain components are printed out as part of Output Item 23.

17. Subroutine MDIN. Subroutine MDIN is called by the main program of Link 4, if mesh point ICN is on the boundary. The flowchart and the source program listing of MDIN are given in Fig. VI-61 and Table VII-70, respectively. The objective of this subroutine is to obtain the direction cosines of the local coordinate axes at mesh point ICN with the specifications described in Sect. VI-E, Volume I; i.e., the first local axis is always normal to the boundary. At the time of entry to the subroutine, matrix DIN contains the direction cosines of the local axes of mesh point ICN, assuming that the mesh point is not on boundary, and vector BIR contains the direction cosines of the outer unit normal vector of the boundary surface at mesh point ICN. In reorienting the local axes, subroutine MDIN calls subroutines UNIT and VECT. There is no error return in the subroutine.

18. Subroutine META. Subroutine META is called by the main program of Link 4. The flowchart and the source program listing of META are given in Fig. VI-62 and Table VII-71, respectively. The objective of this subroutine is to obtain the material matrix (see Fig. III-2b, Vol. I) and thermal expansion coefficients of the current element group associated with ICN/IM/IC, in DD and AL1, AL2, AL3, respectively, in the local coordinate system of mesh point ICN. If the material axes of the current group are not parallel to the local axes of mesh point ICN, subroutine META calls subroutine ROTA to express the material matrix in the local coordinate system. Before returning control to the calling program, subroutine META rearranges the rows and the columns of DD such that the material matrix is arranged with the order of 11, 22, 12, 33, 13, 23. There is no error return in the subroutine.

19. Subroutine QUAD. Subroutine QUAD is called by subroutine DINA if a general shell structure is under question. The flowchart and the source program listing of QUAD are given in Fig. VI-63 and Table VII-72, respectively. The objective of the subroutine is to generate in DIN the direction cosines of the local axes of mesh point ICN of the shell structure, and find the value of ANGLE. The subroutine first obtains in vector MSET the labels of the nodes appearing in the mesh elements corresponding to the current values of ICN/IM/IC. Then it obtains in ZD a vector in the general direction of shell normal (block 10* of Fig. VI-63). After this, the subroutine extracts from vector MSET a list of unrepeatable labels in vector NSET. The order of NSET is in NB. If NB is not smaller than 9, a best-fit quadratic surface passing through mesh point ICN and its immediate neighbors may be possible. If NB is smaller than 9, subroutine QUAD calls subroutine EPAN to enlarge vector NSET and NB to include in the list the labels of the immediate neighbors of the mesh points that are already included in vector NSET, without repetition.

Next, by calling subroutine BEST, subroutine QUAD attempts to generate in vector ZTA the direction cosines of the normal of a best-fit plane (in the least squares sense) to the family of mesh points listed in vector NSET. If this fails, vector ZD is taken as vector ZTA. Then,
assuming that the first local axis is in the direction of vector BAS, the subroutine generates the first approximation of matrix DIN (the first, second, and third columns of matrix DIN are referred to as vectors XII, ETA, and ZTA). If NB is not smaller than 9, subroutine QUAD generates in matrix D the condition equations for a quadratic surface passing through the mesh points listed in vector NSET. The condition equations are obtained in the first approximation of the local axes. These equations are next solved by a least squares method with the help of subroutine INV. If the solution is successful, the local normal is taken as the normal direction of this quadratic surface, and matrix DIN is corrected accordingly. With the use of the new matrix DIN, the process of locating a best-fit quadratic surface is repeated to increase the accuracy. If the process of finding a best-fit quadratic surface fails, the subroutine prints out Error Message 21 and returns control to the calling program with the first approximation of matrix DIN. Otherwise, the subroutine examines the value of IPIR. If IPIR is larger than 1, the subroutine rotates the local axes about the normal until the first local axis is in the smaller principal curvature direction of the best-fit quadratic surface. Initially zero value of ANGLE is changed to the degrees value of the angle between vector BAS and the final orientation of the first local axis. In performing its functions, QUAD also calls subroutines FINDX, SCAL, UNIT, and VECT. There is no error return in the subroutine.

20. Subroutine REVO. Subroutine REVO is called by subroutine DINA if a shell of revolution is under question. The flowchart and the source program listing of REVO are given in Fig. VI-64 and Table VII-73, respectively. The objective of this subroutine is to generate in matrix DIN the direction cosines of the local axes by fitting, if possible, a fourth-order polynomial to the meridional curve. The normal of this curve is taken as the direction for the third local axis. For this purpose, the subroutine first finds the labels of the first four immediate neighbors of mesh point ICN and places them in vector NSET. Vector NSET also contains the label of ICN. The order of NSET is NB. If for some reason NB is less than 5, the subroutine fits a polynomial curve of degree NB-1 to the meridional curve. The conditions for the mesh points listed in vector NSET on the polynomial curve are generated on matrix B and vector C. The unknown coefficients of the polynomial are obtained from these conditions by means of subroutine INV. If the system is singular, and a polynomial curve fit is not possible, the program will use the line segment joining the mesh points confining mesh point ICN as the fitted curve and cause the production of Error Message 22. The normal direction of the fitted curve is taken as the third local axis. The overall Z axis is taken as the negative of the second local axis. The first local axis is tangent to the fitted curve and heads towards the increasing arc distance on the meridian (the meridian curve is assumed directed). The first, second, and third columns of matrix DIN are named as vectors XII, ETA, and ZTA, and contain the direction cosines of the first, second, and third local axes. In obtaining the direction cosines of the local axes, subroutine REVO calls subroutines FINDX, SCAL, UNIT, and VECT.

21. Subroutine ROTA. Subroutine ROTA is called by subroutine META if the local axes are not parallel to the material axes. The flowchart and the source program listing of ROTA are given in Fig. VI-65 and Table VII-74, respectively. The objective of this subroutine is to express the material matrix DD in the local coordinate system defined by matrix DIN (see Table III-5, Vol. II, basic). There is no error return in the subroutine. In obtaining various unit vectors, subroutine ROTA calls subroutines SCAL, CAL, UNIT, and VECT.

22. Subroutine SAME. Subroutine SAME is called by the main program of Link 4 once for every mesh element in the group of current ICN/IM/IC. The flowchart and the source program listing of SAME are given in Fig. VI-66 and Table VII-75, respectively. The objective of this subroutine is to output stresses for the current ICN/IC/IM group, in the local coordinate system if mesh point ICN is a boundary point, and the group is not of shell type. Therefore, this subroutine produces the last portion of Output Item 22. There is no error return in the subroutine.

23. Function SCAL. Function SCAL is called by subroutines BOFI, QUAD, REVO, ROTA, and SETA of Link 4. The flowchart and the source program listing of SCAL are given in Fig. VI-67 and Table VII-76, respectively. The objective of the program is to return to the calling program the scalar product of the vectors referred by the first and second arguments. There is no error return.

24. Subroutine SETA. Subroutine SETA is called by the main program of Link 4 once for every mesh element in the group of current ICN/IM/IC. The flowchart and the source program listing of SETA are given in Fig. VI-68 and Table VII-77, respectively. The objective of the program is to add one additional row to the augmented matrix of strain deflection equations for each direction joining mesh point ICN to the remaining vertices of the mesh element. The subroutine assigns a
weight of 10 or 1 to the equation of a direction, depending upon whether the vertex is on the boundary or not. The weights are recorded in vector IWG. The subroutine generates the row of the augmented matrix as described in Sect. II, Vol. II (basic), by considering thermal strains. When a row is added to the augmented matrix, row count ICON is also updated. In obtaining the thermal strain per unit temperature in a given direction, subroutine SETA calls subroutine TEMP. In achieving various vector operations, it also calls function SCAL, and subroutines FINDQ, FINDX, UNIT, and VECT of Link 4. There is no error return in the subroutine.

25. Subroutine STRA. Subroutine STRA is called by subroutine DIMI. It is identical with subroutine STRA of Link 2. For further information, see Sect. III-B-25. The source program listing of this subroutine is given in Table VII-78. In performing its function, STRA calls subroutine TOPO.

26. Subroutine STRS. Subroutine STRS is called by the main program of Link 4. The flowchart and the source program listing of STRS are given in Fig. VI-69 and Table VII-79, respectively. The objective of this subroutine is to obtain the components of the best-fit stress tensor for current ICN/IM/IC, and list them in vector SR to comply with Table VI-6, Vol. I. At the time of entry to the subroutine, matrix DD contains the material constants, matrix C contains the components of the best-fit usual and angular strains in the first and second columns, respectively, and SR contains the prescribed stresses in the order shown in Table V-2. The subroutine first generates in vector RED the best-fit stresses, then modifies them with the prescribed stresses in SR, and finally copies the final set into vector SR in the order shown in Table VI-8, Vol. I. There is no error return in the subroutine.

27. Subroutine TEMP. Subroutine TEMP is called by subroutine SETA if temperature loading of an anisotropic material is under question. The flowchart and the source program listing of TEMP are given in Fig. VI-70 and Table VII-80, respectively. The objective of this subroutine is to obtain the linear strain in the direction given by the unit vector in XF (see the comment in Table VII-80) due to unit temperature increase, and to store this quantity in the explicit argument. To do this, the subroutine uses matrix W generated by subroutine ROTA and XF generated by subroutine SETA as DCAR. There is no error return in the subroutine.

28. Subroutine TICK. Subroutine TICK is called by the main program of Link 4. It is identical with subroutine TICK of Link 1. For further information, see Sect. II-B-13. The source program listing of this program is given in Table VII-81.

29. Subroutine TOPO. Subroutine TOPO is called by subroutine GENE. It is identical with subroutine TOPO of Link 2. For further information, see Sect. III-B-27. The source program listing of this program is given in Table VII-82.

30. Subroutine TRAN. Subroutine TRAN is called by subroutines DIMI and STRA of Link 4. It is identical with subroutine TRAN of Link 2. For further information, see Sect. III-B-28. The source program listing of the program is given in Table VII-83.

31. Subroutine UNIT. Subroutine UNIT is called by subroutines BEST, BOFI, INLZ, MDIN, QUAD, REVO, ROTA, and SETA of Link 4. The flowchart and the source program listing of UNIT are given in Fig. VI-71 and Table VII-84, respectively. The objective of the subroutine depends upon the contents of the second argument. If the second argument is zero, the subroutine computes the magnitude squared of the vector indicated by the first argument and returns control to the calling program. If the second argument is nonzero, the subroutine replaces the vector in the first argument with a unit vector and the second argument with the magnitude of the original vector. If the second argument, at the beginning, is a positive number, the unit vector is parallel and in the same direction as the original vector. If the second argument, at the beginning, is a negative number, the unit vector is parallel and in the opposite direction of the original vector. There is no error return in the subroutine.

32. Subroutine VECT. Subroutine VECT is called by subroutines BEST, MDIN, QUAD, REVO, ROTA, and SETA of Link 4. The flowchart and the source program listing of VECT are given in Fig. VI-72 and Table VII-85, respectively. The objective of this subroutine is to obtain in the vector indicated by the first argument the cross-product of the vector in the second argument times the vector in the third argument. There is no error return in the subroutine.
VI. Semidetailed Flowcharts

This section contains semidetailed flowcharts of ELAS/Level 3. The flowchart of each program element is treated separately, and given a figure number. The flowcharts are arranged alphabetically by the subroutine names, under the main program of each link. The meanings of the symbols used in the flowcharts may be obtained from the text description of the corresponding subroutine given in the preceding sections and/or Tables III-2 and III-4 of Vol. II (basic). Each flowchart should be considered together with the corresponding source program listing in Sect. VII, and the descriptive paragraph of the earlier sections. The number attached to a block in a flowchart is the number of the first statement in the source program listing corresponding to this block. If the first statement does not have a statement number, the nearest statement number is used with an asterisk in the block. An asterisk before the number in the block means that the first statement of the block is before the statement indicated by the block number. An asterisk after the number in the block means that the first statement of the block is after the statement indicated by the block number. Multiple asterisks indicate qualitatively the distance between the statement with the number and the first statement of the block in the source program listing.
Fig. VI-1. Flowchart of main program of Link 1 (input link)
Fig. VI-1 (contd)
SUBROUTINE ARAN

Initialize procedure

100

ISHUF < 2?

Yes

Read in better labels of nodes sequentially

No

Yes

Yes

No

Error

No

Generate initial IMIN and IMAX vectors using function LEBIN and check ABIN for errors

100*

ISHUF > 0?

Yes

Start measuring relabelling time

No

Error

200*

INF > 1?

Yes

Call subroutine OUTPUT to print out labels, IMIN, IMAX, and the connectivity matrix ABIN

No

100

Prepare for a new sweep

1100* Yes

Find off-bond element count XSA of current sweep

No

Start a new sweep

1100* Yes

Enough non-plain sweeps?

No

Measure time elapsed

1100* Yes

Time to print?

No

1100* Yes

Is this first message?

Yes

1111* Yes

Print message about progress of relabelling

No

Punch out current labelling

1111*

Begin lower sweeping step

Begin upper sweeping step

1111 No

Find change in XSA (IG+IL) if interchange is done

1120

Fig. VI-2. Flowchart of subroutine ARAN (Link 1)
No change in XSA. Compute NZG-NZL to determine if row with more zeros would move outward.

Interchange criteria are satisfied. Interchange current successive rows in ABIN matrix.

Update label vectors ISIR.

Update IMIN and IMAX vectors.

Current sweep complete?

Was any interchange done?

Which is next sweep step?

Call subroutine OUTPUT to print out ISIR, IMIN, IMAX, and connectivity matrix ABIN.

Measure total relabelling time.

Adjust IMAX values so that variable bandwidth is never decreasing.

Print error message.

Punch out ISIR and adjusted IMAX values.

Return.

FIG. VI-2 (contd)
SUBROUTINE COOR

Set error indicator and node count to zero: IERR = 0, L = 0

Set INTE as an even integer that is less than 1001 and minimizes (IN-L-INTE)

Read INTE fields of node label and coordinates. Set i = 0

Set L = IN + 300

Error. Set error indicator to 1

Set I = i + 1

Node label = 0, +?

Node label sequential? Yes

Update node counter L; find storage location of X, Y, Z and store X and Y

No

Node label = 0, +?

Yes

Fig. VI-3. Flowchart of subroutine COOR (Link 1)
SUBROUTINE EXCH

Compute column numbers in ABIN matrix of extremes within bands of rows (and columns) to be interchanged.

Interchange words in that portion of rows that is within the band.

Compute word counts (i.e., column numbers in ABIN) and bit counts of columns corresponding to rows interchanged.

FUNCTION LEBIN (A, I)

Save index register 1 (XRI).

Load accumulator from location I (second argument, which is a FORTRAN integer).

Place complement of I into XRI.

Load accumulator logically from location A (first argument).

Make accumulator zero or nonzero depending upon if Ith bit of A is zero or 1 by ANDing TABLE, I to accumulator.

Accumulator zero?

Yes

No

Load accumulator with FORTRAN integer I.

Restore index register 1 (XRI).

Return.

Fig. VI-4. Flowchart of subroutine EXCH (Link 1)

Fig. VI-5. Flowchart of function LEBIN (Link 1)
SUBROUTINE SEBIN (A, I, N)

Save storage indicators

Save index register I (XRI)

Load storage indicators from location A (first argument)

Load accumulator from location I (second argument, which is a FORTRAN integer)

Place complement of I into XRI

N (third argument) zero? Yes

No

Make ith bit of A in storage indicators 1 by ORing TABLE, I to storage indicators

Make ith bit of A in storage indicators 1 by resetting TABLE, I to storage indicators

Store storage indicators to location A

Re-store index register I (XRI)

Re-store storage indicators

Return

Fig. VI-6. Flowchart of subroutine SEBIN (Link 1)
SET SUBROUTINE MEST

7910

Set element (M), sequence (MT), word (L) counters to zero, minimum word counter (NBE) = -19, maximum word counter (NFI) = 0

7911

Set word counter L = L + 1

Yes

Set word counter L = L + 1

No

Obtain element type number (IELT) as least significant two digits of (L + 1)th word

IELT admissible?

Yes

No

No

Determine number of words necessary for element description and number of vertices

Obtain word count of last descriptive word (UK)

Augment NBE and NFI by 20 and read in 20 more words

Obtain address of UKth descriptive word in COMMON (JM)

Set L = L + 1

Set IERR = 1

Was this last descriptive word in core?

Yes

No

Fig. VI-7. Flowchart of subroutine MEST (Link 1)
Check if all descriptive words are positive

Was this the last element?

NFI > 20?

Shift last-read 20 words back and set NBE - 1, NFI = 20

Set IERR = 0

Error. Set IERR = 1

Return

Fig. VI-7 (contd)
SUBROUTINE OUTPT

-16
Print heading for connectivity information

-17
Print new and old labels, IMIN and IMAX vectors, and (in octal) ABIN matrix

10
Does ABIN have more than 7 columns?

Yes
Print new and old labels and remaining columns of ABIN matrix

No

110

120

Return

Fig. VI-8. Flowchart of subroutine OUTPT (Link 1)
SUBROUTINE SRAT

102

ISHUF > 27

Yes

No

23

25

Generates label vector ISH and corresponding IMAX values

Read in new labels and

Compute column order of

Compute obtained adjusted IMAX values and possible

relabelling

Loop on elements (H) sequentially to generate connectivity matrix ABIN

Obtain descriptive information of Hth element by calling subroutine TOPO

Obtain number of vertices (IMS) of Hth element; check if nodes exist

Expand element vertex vector by those nodes that are related to the vertices by disc input units

Loop on vertices (I) of Hth element, sequentially

Print error message

Loop on vertices (J) of Hth element

Set IB as row of ABIN corresponding to ith vertex, and KN as number of suppressed degrees of freedom of ith vertex

Loop on vertices (J) of Hth element

KN = IDEG?

Yes

No

921

941

Set IB as column bit count of ABIN corresponding to Jth vertex and KM as number of suppressed degrees of freedom of Jth vertex

KM = IDEG?

Yes

No

193

194

1

I = 1

Put 1 as Jth bit of 1 column of 1st row of ABIN matrix

Loop on vertices (J) of Hth element

KN = IDEG?

Yes

No

971

942

Set total number of retained unknowns into ISUM

Compute bandwidth of retained equations of ith node into IBAND vector

Loop on vertices (J) of Hth element

KN = IDEG?

Yes

No

972

99

Loop on I satisfied?

Yes

No

973

99

Loop on M satisfied?

Yes

No

974

9100

Fig. VI-9. Flowchart of subroutine SRAT (Link 1)
SUBROUTINE TABL

Fig. VI-9 (contd)

Fig. VI-10. Flowchart of subroutine TABL (Link 1)
SUBROUTINE TICK (ITIM)

Subroutine is being called first time. Make contents of ONCE nonzero

Subroutine was called before (at time recorded in INITL). Take contents of location 5 logically

Take contents of absolute core location 5 (which contains time in 1/60 second units) logically

Subtract contents of INITL to find elapsed time

Store contents of absolute core location 5 into INITL logically

Make a FORTRAN integer from elapsed time since first call

Store zero to argument ITIM

Return

Return

Fig. VI-11. Flowchart of subroutine TICK (Link 1)
SUBROUTINE TOPO

Set zero to type numbers of descriptive information of Mth element and N(I) block

Find and analyze first descriptive word of Mth element for type numbers

Find and analyze second descriptive word of Mth element for type numbers

Find and analyze third descriptive word (if there is any) of Mth element for type numbers

Find and analyze fourth descriptive word (if there is any) of Mth element for type numbers

Find and analyze fifth descriptive word (if there is any) of Mth element for type numbers

Find and analyze sixth descriptive word (if there is any) of Mth element for type numbers

Obtain and store in N(I) block the node labels of Mth element

1400 Node labels acceptable?

Yes

1410 Print error message

No

1430 Set error counter NDX to zero

1600 Pressure type (ITC) admissible?

Yes

1610 Set pressure type number (IPR) as maximum allowable in descriptive word. Augment error counter NDX by 1

No

1640 Correct ITC as maximum allowable in descriptive word. Augment error counter NDX by 1

1650 Temperature gradient type (USDY) admissible?

Yes

1660 Correct USDY as maximum allowable in descriptive word. Augment error counter NDX by 1

No

1680 Correct IPR as maximum allowable in descriptive word. Augment error counter NDX by 1

1690 Material type (IMT) admissible?

Yes

1620 Correct IMT as maximum allowable in descriptive word. Augment error counter NDX by 1

No

1641 Correct ITEM as maximum allowable in descriptive word. Augment error counter NDX by 1

1670 Correct JSDY as maximum allowable in descriptive word. Augment error counter NDX by 1

Fig. VI-12. Flowchart of subroutine TOPO (Link 1)
Fig. VI-12 (contd)
MAIN PROGRAM OF LINK 2

100

Save print indicator INP

130

Start measuring time by calling subroutine TICK

150

Copy N block to NBO block; find IMS of Mth element; if any negative node labels, set IERR = -2 and branch to 1

170

Compute IDS, and using type number, obtain TE, DT, DG, DGY, DGZ, PRES, M1, M2, M3 values of Mth element

190

Print error message and current values of M, ITTT, IERR

210

Generate UV vector, assuming dilation only

230

Call subroutine STFS to obtain material information select and execute proper subroutine for generation of S, P (may be partially), and modify UV if necessary

250

Loop on elements (M) sequentially to generate reduced stiffness matrix and reduced load vector

270

Restore INP

290

Call subroutine CAS2 to change INP if necessary

310

Set IERR = P(I) = UV(I) = X(I) = Y(I) = Z(I) = X0(I) = Y0(I) = Z0(I); COMMON (200 - 328) = COMMON (200 - 328) = COMMON (200 - 328) = COMMON (200 - 328) = COMMON (200 - 328)

330

Obtain material information of Mth element on D32, E22, D21, E, and G

350

Obtain IMS of Mth element if multiple element

370

Save IMS of Mth element into IMST

390

Call subroutine CUTE to prepare information for (ITTT+1)st subelement of Mth element

410

Set CFE = 0.5

430

Any thermal load No

450

Obtain type numbers of descriptive information and node labels of Mth element by subroutine TOPO

470

Main element stiffness matrix (5), element load vector (P), and area change in P for pressure and acceleration loadings?

490

Clear element stiffness matrix (5), element load vector (P)

500

Clear storage area of reduced stiffness matrix and set ZGEM = IGEM

520

Loop on elements (M) sequentially to generate reduced stiffness matrix Not multiple element Set IERR = 0, and check if Mth element is multiple element

540

Set IERR = 0, and check if Mth element is multiple element

560

Save IMS of Mth element into IMST

580

Call subroutine CUTE to prepare information for (ITTT+1)st subelement of Mth element

600

Set CFE = 0.5

620

Call subroutine CUTE to prepare information for (ITTT+1)st subelement of Mth element

640

Set CFE = 0.5

660

Any thermal load No

680

Obtain type numbers of descriptive information and node labels of Mth element by subroutine TOPO

700

Update (ITTT, 1002, PRCO, M00, X00, Y00, Z00, XU0, YU0, ZU0) for current element or subelement

720

Clear element stiffness matrix (5), element load vector (P) areas

740

Fig. VI-13. Flowchart of main program of Link 2 (generation link)
Fig. VI-13 (contd)
fig. VI-13 (contd)
SUBROUTINE ADM (S, IDS, A, M, IB, JB, C)

Loop on rows (I) of A

Compute row number IS of S corresponding to Ith row of A

Yes Submatrix A on main diagonal of S ?

11 Set J1 = 1

No Set J1 = 3

Loop on columns of A starting from J1th column

Compute column number of S corresponding to Jth column of A

Augment S(IS, JS) and S(JS, IS) of S matrix by A(1, J)*C

(S is stored columnwise as a vector)

90 No

Loop on I Satisfied ?

Yes

10

No Loop on J Satisfied ?

Yes

10*

Return

Fig. VI-14. Flowchart of subroutine ADM (Link 2)
SUBROUTINE BEAM

110

JARE > 0?

Yes

Find and store value of cross-sectional area

115

JMMZ > 0, 1 ?

Yes

Compute address of JMMZ, JMMZ into ICIZJ

120

JMMY > 0?

Yes

Compute upper half of 6x6 planar beam stiffness matrix into A

125

No

Compute address of JMMY into ICIZJ

130

Error. Set IERR = 1

135

Generate symmetric half of A matrix

140

1000

Return
SUBROUTINE CODI

Compute length of element into EL

IF EL > 0 THEN

IF IELT = 37 THEN

Compute direction cosines of element axis as those of local x-axes into DIR(1,1), DIR(1,2), DIR(1,3)

EXIT

IF IJMFI = 0 OR 90 deg THEN

Compute address of angle FI into ICFIJ

IF |FI| > 90 deg THEN

Compute angle FI in radian units

IF FI > 07 THEN

Set SIGNF = -1.

IF DIR(1,1) > 0 THEN

Compute direction cosines of local y-axes accordingly into second row of DIR

IF DIR(1,2) > 0 THEN

Compute direction cosines of local z-axes as cross product of unit vectors on z- and x-axes into third row of DIR

Compute direction cosines of local x-, y, and z-axes into rows of DIR

Error. Set IERR = 1

Return

Fig. VI-16. Flowchart of subroutine CODI (Link 2)
SUBROUTINE CORT

Compute direction cosines of local x-axis as those of line 1-2 into first row of DIR.

Compute direction cosines of local z-axis as those of direction (1-2)x(1-3) into third row of DIR.

Compute direction cosines of local y-axis from those of z- and x-axes into second row of DIR.

Compute coordinates of nodes of element in local axes located at origin of overall coordinate system.

Compute coordinates of nodes 2 and 3 in local axes located at node 1.

Return

Fig. VI-17. Flowchart of subroutine CORT (Link 2)
SUBROUTINE CUTE (ITTM)

100
Set NAV = NAV + 1, KBAS = KNOO = 0

110
Set IELT = 100

120
Set ITTM = 5, IMS = 4, JEN = 20, IEN = 1

130
Set KBAS = 4

140
Set IELT = IELT - 1, IDS = MS*IDEG

150
Loop on J to copy labels of necessary nodes from N block to NOO block

160
Find sequence number in NOD block

170
If JEN 1 or 2?

180
Quadrilateral case: Compute sequence number of node in block N

190
Hexahedral case: Compute sequence number of node in block N

200
Return

Fig. VI-18. Flowchart of subroutine CUTE (Link 2)
SUBROUTINE DARN (KS, KBS, CCC, KQE)

Compute KS-th entry of IBB vector into KB

*302

KB + IND < 0?

No

Yes

Set ICOMP = 10,000 + KS, INCR = 1

*301

Loop on rows (ISOR) of IBO vector

*302

ISOR-th element absolute value of IBO vector = ICOMP

No

Yes

Set INCR-th elements of KBS and CCC vectors as absolute value of ISOR-th element of IBB and of ISORTH element of C vectors, respectively, and increase INCR by 1.

*306

Loop on ISOR satisfied?

No

Yes

Set KQE = INCR - 1

*308

Return

*301

Set KQE = 1, CCC(1) = 1, KBS(1) = -KB

*304

Set KQE = 1, KBS(1) = KB, CCC(1) = KB-th element of C vector

*309

K$\neq$ 0, +?

Yes

No

Fig. VI-19. Flowchart of subroutine DARN (Link 2)
SUBROUTINE DMM(A, B, M, C)

Loop on rows of matrix A

Set ISS = I - M

Loop on columns of matrix A

Compute sequence number of (I,K)th element of matrix A in one-dimensional form into ISS

Increase C(I) by A(I,K)*B(K) into EL

Loop on K

N EL > 0?

No

Yes

Compute direction cosines of elements of vector T

Yes

Return

Fig. VI-20. Flowchart of subroutine DMM (Link 2)

SUBROUTINE ELDI

Compute length of element into EL

EL > 0?

Yes

Error. Set ERR = 1

No

Compute direction cosines of element axis and state into first, initial, and final elements of vector T

Yes

Return

Fig. VI-21. Flowchart of subroutine ELDI (Link 2)
SUBROUTINE PLBE

*130 JMMX > 0?

No
Yes

Compute address of torsional constant into ICIXJ

150

JMMY > 0?

Yes
No

Compute address of y-moment of inertia into ICIVJ

190

Compute upper half of 6 x 6 stiffness matrix of grid beam into A

260

Generate symmetric half of matrix A

1000

Return

Fig. VI-22. Flowchart of subroutine PLBE (Link 2)

SUBROUTINE RLOC

Compute final column and raw numbers (JJH and IIN) of A, and constants IRE and L for location on S

100

Loop on columns (J) of A from JJ to JJN

260

Set L = L + I05

100

Loop on rows (I) of A from II TO IIN

260

Compute location (LR) in S; copy A(I, J) onto S(LR) and set A(I, J) = 0

100

Error. Set IERR = 1

Fig. VI-23. Flowchart of subroutine RLOC (Link 2)
SUBROUTINE S01

Set IPBG = 0

Prepare for generation of element stiffness matrix (S) in overall coordinate system

Loop on columns (I) of element stiffness matrix

Compute address of cross-sectional area into ICARJ, and set wind direction into DUG block

Call subroutine ELDI to generate pressure direction PD, member axis direction T(D), T(D), and PN = T(D)

Loop on rows (J) of PD, member axis direction element stiffness matrix T(1), T(3), T(5), and PN = T(D)

Compute location of current element of stiffness matrix and generate it

Prepare for contribution of pressure to element load vector by setting PRCO, IPBG, and IPEN values

Prepare for contribution of acceleration to element load vector by setting CONS, IPBG, and IPEN

Prepare for contribution of load to element load vector by setting CONS, IPBG, and IPEN

Error. Set IERR = 1

Loop on I satisfied?

Yes

ACEL = 0?

No

Yes

Prepare for contribution of load to element load vector by setting CONS, IPBG, and IPEN

Prepare for contribution of pressure to element load vector by setting PRCO, IPBG, and IPEN

Prepare for contribution of acceleration to element load vector by setting CONS, IPBG, and IPEN

Loop on J satisfied?

Yes

JSPR = 0?

No

Satisfied?

Yes

Return

Fig. VI-24. Flowchart of subroutine S01 (Link 2)
SUBROUTINE S02

Call subroutine CODI to generate direction cosines of element local coordinate axes in DIR

Call subroutine BEAM to generate planar-beam stiffness matrix in local coordinates in matrix A

Set IPBG = 0, which implies no load-vector generation in main program for pressure and acceleration loadings

Yes

JPRS = 0 ?

No

Set vector DUG as unit vector of first overall coordinate axis

Call subroutine ELDI to generate transverse direction for acceleration in PD, member-axis direction in T, and PN = T/PD

Compute CONS constant and fixed-end moments due to acceleration loading

Set IPBG and IPEN for computation in main program of rest of acceleration-load vector

Initialize and call subroutine RLOC to copy stiffness matrix from A into S, columnwise

By calling subroutine STRA, express element stiffness matrix (S) in overall coordinates

Generate thermal-end rotations and complete thermal-distortions vector of free-free element in UV vector

Return

Fig. VI-25. Flowchart of subroutine S02 (Link 2)
SUBROUTINE S03

Call subroutine CODI to generate direction cosines of element local coordinate units in DIR.

Call subroutine PLBE to generate grid-element stiffness matrix in local coordinates in matrix A.

Set IPBG = 0 to skip pressure and acceleration load vector generation in main program. Set IPEN = 1.

By calling subroutine RLOC, generate direction cosines copy stiffness matrix in A into S, columnwise.

Call subroutine PLBE to Yes DGZ = 0? No

Generate vector UVG for thermal rotations of free-free beam in local coordinates.

Generate vector UVG for thermal distortions of free-free beam in UV vector by using UVG.

By calling subroutine TRAN, express vector UVG in overall coordinates.

Generate thermal distortions of free-free beam in overall coordinates.

By calling subroutine STRA, express element stiffness matrix S in overall coordinates.

Set IPEN = 2. Return.

Compute force magnitude at ends due to pressure into PRCO.

Compute element-load vector in overall coordinates due to PRCO.

JPRS = 0?

IPEN 1 or 2?

ACEL = 0?

Compute force at ends due to acceleration into PRCO. Set IPEN = 2.

Fig. VI-26. Flowchart of subroutine S03 (Link 2)
SUBROUTINE S04

Call subroutine CODI to generate direction cosines of element local coordinate axes in DIR

Set IPBG = 0, which implies no load-vector generation in main program for pressure and acceleration

If IPBG = 0, call subroutine ELDI to generate transverse direction in PD, member-axis direction in T, and PPD = 0.5

Compute constant CONS and fixed-end moments due to acceleration loading

Set vector DUG as unit vector of first overall coordinate axis

Call subroutine ELDI to generate transverse direction for acceleration-load vector in main program for pressure

Compute thermal-end rotations in free-free beam because of DGY into UVG

Set vector DUG as unit vector of first overall coordinate axis

Call subroutine ELDI to generate transverse direction for acceleration-load vector in main program for pressure

Compute thermal-end rotations in free-free beam because of DGY into UVG

Compute thermal-end rotations in free-free beam due to DGZ into UVG

Fig. VI-27. Flowchart of subroutine S04 (Link 2)
SUBROUTINE S05

Compute twice the area of triangle into A2

Yes

A2 > 0?

By calling subroutine TRM, generate \([E]_{11}\) \([E]_{22}\) \([E]_{33}\) \([E]_{12}\) \([E]_{21}\) \([E]_{13}\) \([E]_{31}\) \([E]_{23}\) \([E]_{32}\)

Yes

Generate \(F1\), and by means of subroutine ADM, add \(F1\) multiple of \([G]\) submatrices on element stiffness matrix \(S\), starting at location \((1,1)\)

Yes

By calling subroutine TRM, generate \([E]_{1}^T \[E]_{2}\) \([E]_{3}\) \([E]_{4}\) into \([E]M\)

No

By means of subroutine ADM, add \(F1\) multiple of \([G]\) submatrices on element stiffness matrix \(S\), starting at location \((4,4)\)

No

By calling subroutine TRM, generate \([E]_{11}\) \([E]_{22}\) \([E]_{33}\) \([E]_{12}\) \([E]_{21}\) \([E]_{13}\) \([E]_{31}\) \([E]_{23}\) \([E]_{32}\)

Yes

ACEL > 0?

Set IPBG = 0 to skip pressure and acceleration load vector generation in main program

Set IPBG = 0 to skip pressure and acceleration load vector generation in main program

Compute contribution of pressure on side of triangle into \(P\), considering CFE value

Yes

No

Set KAV = HAV

Yes

No

Set KAV > 2?

Yes

No

Compute contribution of pressure on side of triangle into \(P\), considering CFE value

Yes

No

Error: Set IERR = 1

Set PRCO = 0.

Compute CONS, IPBG, and IPEN values so that acceleration loads are computed in main program.

Return

By calling subroutine TRM, generate \([E]_{1}\) \([E]_{2}\) \([E]_{3}\) \([E]_{4}\) into \([E]M\)

By calling subroutine TRM, generate \([E]_{11}\) \([E]_{22}\) \([E]_{33}\) \([E]_{12}\) \([E]_{21}\) \([E]_{13}\) \([E]_{31}\) \([E]_{23}\) \([E]_{32}\)

Yes

No

Set IERR = 1

Return

Fig. VI-28. Flowchart of subroutine S05 (Link 2)
SUBROUTINE S07

Compute twice the area of triangle into $A_2$

$A_2 > 0$?

Yes

No

By calling subroutine TRM, generate $[EM], [EN], [EQ]$ submatrices. Compute F1 constant

By calling subroutine TRM, generate $[EQ], [EM], [EQ]$ submatrices.

Examine $[EQ]$ to see if triangle is obtuse; if so, apply constant-trace scheme to modify $F_2$; if not, continue to next operation

By means of subroutine ADM, add $F_2$ multiple of submatrix into $[EQ]$, on element stiffness matrix $S$, starting at location $(4,4)

By calling subroutine TRM, generate $[EQ], [EQ], [EQ]$ submatrices. Compute constant $F_2$

By calling subroutine ADM, add $F_2$ multiple of $[EM]$ on element stiffness matrix $S$, starting at location $(4,4)$ and change sign of $F_2$

Set PRCO = 0 to skip pressure-load computation in main program.

Apply equilibrium algorithm into P vector to generate the part of $S$ associated with submatrix $Q$

By calling subroutine TRM, generate $[EM], [EQ], [EQ]$ into $[EQ]$. Compute constant F2

By means of subroutine ADM, add $F_2$ multiple of submatrices into $[EQ]$, on element stiffness matrix $S$, starting at location $(7,7)$.

By means of subroutine ADM, add $F_2$ multiple of submatrices into $[EQ]$, on element stiffness matrix $S$, starting at location $(4,4)$

Return

By calling subroutine TRM, generate $[EQ], [EQ], [EQ]$ into $[EQ]$

By means of subroutine ADM, add $F_2$ multiple of submatrix into $[EQ]$, on element stiffness matrix $S$, starting at location $(4,4)$

By calling subroutine TRM, generate $[EM], [EQ], [EQ]$ into $[EQ]$

By calling subroutine ADM, add $F_2$ multiple of submatrix into $[EQ]$. Compute constant $F_2$

By calling subroutine TRM, generate $[EQ], [EQ], [EQ]$ submatrices.

Examine $(4)$ to see if triangle is obtuse; if so, apply constant-trace scheme to modify $F_2$; if not, continue to next operation

By means of subroutine ADM, add $F_2$ multiple of $[EM]$ on element stiffness matrix $S$, starting at location $(1,1)$

Compute pressure loading into P vector

By calling subroutine TRM, generate $[EM], [EQ], [EQ]$ into $[EQ]$. Compute constant F2

By means of subroutine ADM, add $F_2$ multiple of submatrices into $[EQ]$, on element stiffness matrix $S$, starting at location $(7,7)$

By means of subroutine ADM, add $F_2$ multiple of submatrices into $[EQ]$. Compute constant $F_2$

By means of subroutine ADM, add $F_2$ multiple of submatrix into $[EQ]$. Compute constant $F_2$

Return

Yes

$DG = 0$?

No

Compute and store thermal-distortion vector of free-free element into UV block

Error. Set ERR = 1

Fig. VI-29. Flowchart of subroutine S07 (Link 2)
SUBROUTINE S09

Compute 6 times the volume of element into DET

If DET > 0?

No

Set VOL=1/(6*DET)

Generate 3x4 geometric matrix into AT

Initialize IPBG by setting IPBG = 0

Yes

JMES = 0?

No

0 < ITTT < 3?

No

Yes

Set IPBG, IPEN, and CONS values for computation of P in main program

Yes

Compute E1^2 (line 1-2) into IX as reference volume

DET > IX?

Yes

This element is too small to consider. Print message and skip operation on this element

No

Loop on J satisfied?

No

Loop on I satisfied?

No

Error. Set IERR = 1

Return

Fig. VI-30. Flowchart of subroutine S09 (Link 2)
SUBROUTINE S11

Call subroutine CORT to express coordinates of element nodes in local coordinates in X, Y, Z, XD, YD, and ZD blocks

Call subroutine S05 to compute membrane stiffness matrix in local coordinates and add it to element stiffness matrix S

Call subroutine S07 to generate bending and transverse shear stiffness matrix and load vector in local coordinates in S and P blocks

Update IPEN so that acceleration loading is extended to z-direction in main program

S07 assumed that IDS = 18. Shift stiffness matrix and load vector in S and P blocks for shell

S07 also generated UV vector for IDS = 18. Shift information in UV to shell

Complete thermal-distortions vector UV by in-plane expansions of free-free element

Return

Fig. VI-31. Flowchart of subroutine S11 (Link 2)

SUBROUTINE S13

Call subroutine CORT to express coordinates of element nodes in local coordinates in X, Y, Z, XD, YD, and ZD blocks

Shift pressure loading in P generated by S07 with IDS = 18 properly for shell

Generate P vector for pressure loading and set PRCO = 0 for no pressure load computation in main program

Set IPEN=3 so that acceleration loading is extended to z-direction in main program

Save IGEM into IGIE and set IGEM=0 so that subroutines TRAN and STRA are used in matrix transformation of next block

Call subroutine TRAN for P vector and STRA for S matrix for representations in overall coordinates

Restore IGEM value

Return

Fig. VI-32. Flowchart of subroutine S13 (Link 2)
SUBROUTINE S15

Set error bounds for X and \( X_{1+n} \), respectively.

Compute abscissa of centroid of triangle into XBAR. Save contents of TE into TBR, and set TE = XBAR.

Save material matrix in D21 into E21.

Generate D33 matrix from E21.

By calling subroutine S50, generate stiffness matrix of triangle into S with thickness XBAR.

By calling subroutine 505, generate stiffness matrix of triangle into S with thickness XBAR.

Compute ordinate of centroid of triangle into YBAR, and set YBAR = 0.

Set material vector into third column of E22.

Set first and second columns of E22 as XBAR and YBAR multiples of third column.

Set \( F_1 = 1/(2*A_2) \), where \( A_2 \) is generated by subroutine S50 (2A is dropped from element matrices).

Set third row of EQ matrix as third column of X.

Set V vector as y-coordinates of nodes about axis passing through centroid.

Clear area of Q.

Set IAX as sequence number of last off-Y-axis node; generate D33 matrix from NN as total number of nodes on Y-axis with ERI tolerance; and LL as sequence number of last on-Y-axis node. Check if abscissas are negative.

By calling subroutine 505, generate stiffness matrix of triangle into S with thickness XBAR.

Negative abscissa exists. Error. Set \( IERR = 1 \).

By calling subroutine TRIM, generate EM and EN submatrices.

Set indicator MM to zero.

Loop on sides (L) of triangle.

Set sequence number of second end into M.

Set \( X_{1+m} \) and \( Y_{1+m} \) into JXX and JYY, respectively.

Fig. VI-33. Flowchart of subroutine S15 (Link 2)
Fig. VI-33 (contd)
Modify \([Q]\) properly by using CAX vector for the case of "2 nodes on Y-axis" (no modification if this is not the case).

By calling subroutine ADM, add \(F1\) multiple of \([Q]\) on element stiffness matrix \(S\), starting at location (1, 1).

By calling subroutine TRM, generate \([\text{EN}]^T[\text{CG}]\) (second term of \(S\)) into \([\text{EM}]\).

Add transpose of \([\text{EN}]\) to itself (\([\text{EN}]^2\)).

By means of subroutine ADM, add \(F1\) multiple of \([\text{EN}]\) on element stiffness matrix \(S\), starting at location (1, 1).

Restore \(K\) from \(A\), and compute \(\text{IER}\) and \(\text{CONS}\) quantities for acceleration loading.

Generate auxiliary quantities on first column of \([Q]\), and using them with \([\text{EQ}]\), \([X]\) T matrix, generate three quantities on second column of \([Q]\).

Using second column of \([Q]\), CONS value, and \(G1\), \(G2\), obtain load vector due to acceleration.

Consider the CFE value, compute and increment load vector by pressure loading.

Set \(\text{IPSG} = -1\) to skip acceleration- and pressure-loading computation in main program.

Restore material constants from \(221\) into \(231\).

Return.

Fig. VI-33 (contd)
SUBROUTINE S17

Determine length and direction of midpoint of segment into BOY and XAV

1

N

Yes

*1

Define [Q] = [BE/2
B/2]

Define Fl = TE*D33(2, 2)

Compute direction cosines of 1-2 direction into AL and BE and set
GA = BOY * XAV

Set first row of [Q] as
[AL AL -BE BE]

Fig. VI-34. Flowchart of subroutine S17 (Link 2)
By calling subroutine ADM, add F1 multiple of 2x2 stiffness matrix, starting at location (1, 1).

Set IPBG = -1 to skip computation of pressure and acceleration load in main program.

Compute auxiliary quantities for load-vector computation into AL, BE, FI, and F2.

Using AL, BE, F1, F2, generate load vector due to pressure and acceleration loading on P.

Return.

Fig. VI-34 (contd)
SUBROUTINE S18

Call subroutine S17 to generate part of stiffness and load matrices related with membrane action.

Define $F_2 = \mathbf{T}E^2/2$

If Cylindrical segment?

Yes

Any of the ends on Y-axi?

Yes

Set $F_1 = F_2*D_33(1,1)*XAV(1)/BOY$

Define $F_1 = -F_1$

By calling subroutine ADM, add $F_1$ multiple of $2\times2$ matrix on element stiffness matrix, starting at location $(5,5)$.

By calling subroutine TRM, generate first row of EM on element stiffness matrix, starting at location $(1,1)$.

Set $F_1 = F_2*D_33(1,1)*XAV/BOY$

No

If $X(2) = 0$?

Yes

Define $F_1 = F_2*D_33(1,1)*XAV(1)/BOY^2$

No

Define $F_1 = F_2*D_33(2,1)*XV(1)/BOY$

No

By calling subroutine ADM, add $F_1$ multiple of $2\times2$ matrix on element stiffness matrix, starting at location $(5,5)$.

By calling subroutine ADM, add $F_1$ multiple of $F_2$ as part of transverse shear stiffness matrix on element stiffness $S$, starting at location $(5,5)$.

Set $[Q] = (BOY)^2/2$ [$1 \quad 1$]

$43^*$

Define $F_1 = F_2*D_33(1,1)*XV(1)/BOY$

$45^*$

Define $F_1 = F_2*D_33(2,1)*XV(1)/BOY$

$47^*$

Define $F_1 = F_2*D_33(1,1)*XV(1)/BOY$

$49^*$

Define $F_1 = F_2*D_33(2,1)*XV(1)/BOY$

$51^*$

Define first row of EM on element stiffness matrix, starting at location $(1,1)$.

By calling subroutine ADM, add $F_1$ multiple of $[G]$ and $[G]^T$ as part of transverse shear stiffness matrix on element stiffness $S$, starting at location $(5,5)$.

Define $[Q] = (G)S$.

$55^*$

By calling subroutine ADM, add $F_1$ multiple of $[G]$ and $[G]^T$ as part of transverse shear stiffness matrix on element stiffness $S$, starting at location $(5,5)$.

$57^*$

Complete thermal distortions vector $U[V]$ by thermal rotations of free-free element.

Return

Fig. VI-35. Flowchart of subroutine S16 (Link 2)
SUBROUTINE STFS (IELT)

IELT = ?

Call subroutine S01 to generate element matrices

IELT = 1

Call subroutine S02 to generate element matrices

IELT = 2

Call subroutine S03 to generate element matrices

IELT = 3

Call subroutine S04 to generate element matrices

IELT = 4

Call subroutine S05 to generate element matrices

IELT = 5

Call subroutine S06 to generate element matrices

IELT = 6

Call subroutine S07 to generate element matrices

IELT = 7

Call subroutine S08 to generate element matrices

IELT = 8

Call subroutine S09 to generate element matrices

IELT = 9

Call subroutine S10 to generate element matrices

IELT = 10

Call subroutine S11 to generate element matrices

IELT = 11

Call subroutine S12 to generate element matrices

IELT = 12

Call subroutine S13 to generate element matrices

IELT = 13

Call subroutine S14 to generate element matrices

IELT = 14

Call subroutine S15 to generate element matrices

IELT = 15

Call subroutine S16 to generate element matrices

IELT = 16

Call subroutine S17 to generate element matrices

IELT = 17

Call subroutine S18 to generate element matrices

IELT = 18

Return

Fig. VI-36. Flowchart of subroutine STFS (Link 2)
SUBROUTINE STRA

Prepare to express columns of $S$ in overall coordinates by setting $J = -IDS$ ($S$ is an $IDS 	imes IDS$ square matrix)

Loop on columns ($i$) of $S$

Compute pointer (relative to $S(1,1)$ of $i$th column of $S$ into $J$

By calling subroutine TRAN, express $i$th column of $S$ in overall coordinates

Loop on $I$ satisfied?

Prepare to transpose matrix $S$ by setting IABB = -IDS and IBBB = -IDS

Loop on rows ($i$) of matrix $S$

Compute pointers of elements in first row of $S$ into IAB and IBE

Loop on columns ($J$) of matrix $S$

Compute element count of $S(i, J)$ into IAB, and element count of $S(J, 1)$ into IBE

Is IAB = IBE?

Interchange $S(i, J)$ and $S(J, i)$

Loop on $J$ satisfied?

Loop on $I$ satisfied?

Prepare to express columns of $S$ in overall coordinates by setting $J = -IDS$

Loop on columns ($i$) of $S$

Compute pointer (relative to $S(1,1)$ of $i$th column of $S$ into $J$

By calling subroutine TRAN, express $i$th column of $S$ in overall coordinates

Loop on columns ($J$) of $S$

Compute element count of $S(i, J)$ into IAB, and element count of $S(J, 1)$ into IBE

Loop on $I$ satisfied?

Return

Fig. VI-37. Flowchart of subroutine STRA (Link 2)
SUBROUTINE TOPO

10

Set zero to type numbers of descriptive information of Mth element and \( N(I) \) block

100

Find and analyze first descriptive word of Mth element for type numbers

200

Find and analyze second descriptive word of Mth element for type numbers

300

Find and analyze third descriptive word (if there is any) of Mth element for type numbers

400

Find and analyze fourth descriptive word (if there is any) of Mth element for type numbers

500

Find and analyze fifth descriptive word (if there is any) of Mth element for type numbers

600

Find and analyze sixth descriptive word (if there is any) of Mth element for type numbers

800

Obtain and store in \( N(I) \) block node labels of Mth element

900

Node labels acceptable?

Yes

1000

Print error message

1100

Return

No third word

No fourth word

No fifth word

No sixth word

Fig. VI-38. Flowchart of subroutine TOPO (Link 2)
SUBROUTINE TRIM (A, IFS)

10 Initialize subvector number
   IGEMP = GEM + 1; pointer
   relative to DUM(I), LK = IFS + 4*IMAS

20 Loop on subvector count L
   (L = 1 means displacements subvector; L = 2 means
   rotations subvector)

30* Compute pointer relative to first word of Lth subvector
   into LK

40 Loop on columns (J) of DIR
   (J = 1 first, J = 2 second, and J = 3 third
   overall axis)

50* Loop on vertices (I) of element

60 Compute count of corresponding element into
   LK and clear DUM (LJK)

70 Compute pointer of element in A into LKI

80 Loop on rows (K) of DIR
   (K = 1 first, K = 2 second, K = 3 third
   overall axis)

90* Compute count of element in vector A into LKI

100* Add contribution of A(LKI)
    on DUM(LJK)

110 Return

Fig. VI-39. Flowchart of subroutine TRAN (Link 2)

SUBROUTINE TRAN (A, IFS)

10 Normalize subvector number
   IGEMP = GEM + 1; pointer
   relative to DUM(I), LK = IFS + 4*IMAS

20 Loop on subvector count L
   (L = 1 means displacements subvector; L = 2 means
   rotations subvector)

30* Compute pointer relative to first word of Lth subvector
   into LK

40 Loop on columns (J) of DIR
   (J = 1 first, J = 2 second, and J = 3 third
   overall axis)

50* Loop on vertices (I) of element

60 Compute count of corresponding element into
   LK and clear DUM (LJK)

70 Compute pointer of element in A into LKI

80 Loop on rows (K) of DIR
   (K = 1 first, K = 2 second, K = 3 third
   overall axis)

90* Compute count of element in vector A into LKI

100* Add contribution of A(LKI)
    on DUM(LJK)

110 Return

Fig. VI-40. Flowchart of subroutine TRIM (Link 2)

SUBROUTINE TRM (A, B, C, M, NI)

12 Set N = NI

20 Loop on vertices (I)

30* Compute count of corresponding element into
   LK and clear DUM (LJK)

40 Loop on subvector count L
   (L = 1 means displacements subvector; L = 2 means
   rotations subvector)

50* Compute pointer relative to first word of Lth subvector
   into LK

60 Loop on columns (J) of DIR
   (J = 1 first, J = 2 second, and J = 3 third
   overall axis)

70* Loop on vertices (I) of element

80 Compute count of corresponding element into
   LK and clear DUM (LJK)

90* Compute pointer of element in A into LKI

100 Loop on rows (K) of DIR
   (K = 1 first, K = 2 second, K = 3 third
   overall axis)

110 Compute count of element in vector A into LKI

120* Add contribution of A(LKI)
    on DUM(LJK)

130 Return

Fig. VI-41. Flowchart of subroutine TRM (Link 2)
Fig. VI-42. Flowchart of main program of Link 3 (deflection link)
SUBROUTINE ELST

1. Compute pointer for element set information in COMMON into ISN; set number of words for this purpose, assigned to each node, into IELM; set IELM = IELM + 1

2. Compute first and last word counts of element set information in COMMON and clear this area; set IONE = INCT = 0

3. Loop on vertices (J) - find Ith element

4. Find type number of Ith element into IELT

5. Using IELT, see if element is one-dimensional; if not, set in KN the number of vertices, and in KJ the count of the word, in element descriptive information, which contains label of first vertex

6. Loop on elements (I)

7. Find type number of Ith element into IELT

8. Using IELT, see if element is one-dimensional; increment one-dimensional element count by 1; IONE = IONE + 1

9. Loop on J satisfied?

10. Update element count of NNth node by IE, and store label of this element or (IE+1)st word of element set information of NNth node

11. Print message that there are too many elements at this node for allocated space in COMMON and set indicator INCT to 1

12. Print message that there are too many elements at this node for allocated space in COMMON and set indicator INCT to 1

Fig. VI-43. Flowchart of subroutine ELST (Link 3)
Find first word address into IEN and its contents into IE of element set information of Ith node.

If IE > 0

Copy labels of IE elements meeting at Ith node into IMNT vector from COMMON.

Write on tape ITAS the node label L, number of elements IE, and labels of these elements IMNT.

Loop on nodes (I)

No

Set IFAS = 0 and print error message.

Yes

Return

Fig. VI-43 (contd)
Fig. VI-44. Flowchart of subroutine RESI (Link 3)
SUBROUTINE RESW

Print out heading for residual forces

Clear an area of six words for residual force components in P vector

Loop on nodes (I)

Compute pointer of first word of residual forces of Ith node into IST1

Loop on degrees of freedom (J). Compute address of Jth residual force component into ISTJ. Set L = J

Set L = J + 2

Set L = J + 3

Copy Jth component of residual force of Ith node into P(L)

No Loop on J satisfied?

Yes

Yes

Return

Fig. VI-45. Flowchart of subroutine RESW (Link 3)
SUBROUTINE VELAS (NN, IERR, IST, IDEF)

Copy pointer of diagonal element count to IST and set IST = IST + 1 and Z = 1, E-16

NN > 0 ?

1000

Set N = NN (number of unknowns), N1 = N + 1

1001

First select diagonal element in magnitude and store in E

Using E, recompute allowable minimum for diagonal elements in Z

110

First diagonal element larger than E ?

15

N = 1 ?

1092

Obtain U(I, J) on A(I, J) by A(IST+1) = A(IST+1) - A(IST+J)/A(IST+1)

108

Compute number of elements in first row into IW

1101

IW > 1 ?

1223

Compute remaining of first row of U on corresponding locations of first row of A

121

Compute address of first word following matrix A into MAX (N words following A are for element counts in each column of A)

701

Clear N words following matrix A. Prepare to obtain rest of U on A

701*

Loop on remaining rows (i) of matrix A

201*

Compute column number of first element of ith row into JMX, address of A(I, I) into IO, and set IJ = IO + 1

1072

Loop on columns (J) from I to IW to obtain elements of U in i-th row

2072

IW < 1 ?

16

Increase element count of columns from 1 to IW (IW = JMX of previous row)

119

Loop on K No

101

Loop on K satisfied ?

1101

Compute address of A(K, J) into JK, and address of A(1, I) into IK

1042

Obtain contributions of A(K, J) and A(I, J) to U(I, J) by A(I, J) = A(IST+I) - A(K, J) * A(I, J)

111

Loop on K

102

Loop on row (K) of A from KB to IEF

1402

Compute address of A(K, J) into JK, and address of A(1, I) into IK

121

Obtain contributions of A(K, J) and A(I, J) to U(I, J) by A(I, J) = A(IST+I) - A(K, J) * A(I, J)

100

Loop on K satisfied ?

1101

Compute U(I, J) by A(I, J) on A(I, J)

122

Loop on J satisfied ?

11

Loop on I satisfied ?

701*

Fig. VI-46. Flowchart of subroutine VELAS (Link 3)
Set IM as address of last diagonal element

Start forward sweep by computing \( Y(1) \) on \( B(1) \). Prepare to obtain rest of vector \( Y \)

Loop on rows \( I \) of \( U \) from 2 to \( N \)

Start backward sweep by first computing \( X(N) \) on \( Y(N) \). Prepare to obtain rest of vector \( X \)

Loop on rows \( L \) of \( U \) from \( N-1 \) to 1

Compute \( U(K,1) \) and \( Y(K) \) into \( IK \) and \( JK \)

Obtain contributions of \( U(K,1) \) and \( Y(K) \) to \( Y(L) \) by \( Y(L) = Y(L) - U(K,1)*Y(K) \)

Loop on \( K \) satisfied?

Return

Fig. VI-46 (contd)
Fig. VI-47. Flowchart of main program of Link 4 (stress link)
Determine class number ICLA of ICh class group of IMth material group at node ICN.

Call subroutine IM1Z to initialize vectors BAS and NEX, matrices DIN and W, and constants ICON, ANGLE, IROT, BST, IE, DT, DG.

ICAS < 5 ?

Shell type. Set IROT = 1 and BST = 2H**

Call subroutine DNA to generate DIN matrix, BAS vector, and constant angle.

INBON = 0 ?

No

Call subroutine MDIN to rotate local axes at node ICN so that KSI axis is in ZTA-BIR plane.

No

Determine column number IE of last element in ICh class group of IMth material group at node ICN.

Loop on columns (IL) of MAC matrix (for IM, IC prescribed) from 2 to IE.

INP < 2 ?

No

Print heading for strain deflection equations along nodal lines.

Obtain element label IELT of (IL-)th element of ICh class group of IMth material group at node ICN.

By calling subroutine SETA, obtain strain deflection equations in A for those nodal lines of element IELT that pass through node ICN.

Clear area A in FF for strain deflection equations.

Fig. VI-47 (contd)
Yes

IROT = 0

Print out direction cosines of local axes KSL, ETA, and ZTA, and deflections of node ICN in local coordinates

No

By calling subroutine LEST, solve for unknown strains at node ICN for ICth class group of IMth material group by least squares

$\text{IERR} = 0$

Yes

No

Call subroutine STRS to compute stresses at node ICN for ICth class group of IMth material group in SR vector

Print out stresses in SR vector by indicating node, material, class coordinate system, and boundary relation

Yes

$\text{INP} < 2$

No

Skip two lines in output to separate stress computation of next group

$\text{IERR} = 0$

Yes

Call subroutine DIMI to compute and print stress resultants at ends of one-dimensional elements

No

$\text{IONE} > 0$

Yes

Call subroutine TICK to measure elapsed time in Link 4. Print elapsed time

No

Call Link 1

Fig. VI-47 (contd)
SUBROUTINE ABEQ

Copy residual vector of node ICN into RES from COMMON

Initialize IEQ, IREB, IREN, CT, CR, CL constants and vector M such that their final values are obtained with least modification

Initialize RED and NRED matrices for general case

First vertex on Y-axis?

Set XX = 1.

Set XX = XN[J]

Set equation count ICON as zero

Modify initialized quantities in blocks 9* and 9** above, according to class type

Loop on number of right-hand sides (J) of stress boundary condition equations

Set zero into six elements of vector RED

Copy proper components of residue vector RES of node ICN into RED vector

Express the first three components of residue vector RED in local coordinate system in vector SIR

Loop on stress boundary condition equations (I)

Obtain column number of right-hand side in II, related component number of SIR in K, and related column number of material matrix in LL

Obtain jth right-hand side of Ith equation

Obtain coefficients of unknown strains in Ith equation and set weight of this equation as 100

Loop on J specified?

Modify IEB, IREN, CR, and ICON for second right-hand side

Take from each of IEQ equations power right-hand side, modify it suitably, and store successively in SR vector

Loop on J specified?

Loop on number of right-hand sides (J)

Obtain column number of Jth right-hand side in K

ICAS = 2?

Set CL = CL/CT

Loop on J specified?

Modify initialized quantities satisfied successively

Print out equations for stress boundary conditions, their weights, and prescribed stresses

Return

Fig. VI-48. Flowchart of subroutine ABEQ (Link 4)
SUBROUTINE BEST (CIR, JBAN, MZ)

Set equation count L to zero

Loop on neighboring nodes (0 from 1 to MZ)

Obtain label of 1st neighboring node in K, and its overall coordinates in Xf vector.

Set L = L + 1

Obtain condition (in a coordinate system parallel to overall and originating at XN(1-1.15, XN(2)-1.16, XN(3)-1.17) that ith neighboring node is on the plane, as the Lth equation.

Loop on L satisfied?

No

Yes

Set DET = 0

*16

Are there less than three equations?

No

Yes

By premultiplying matrix in A by transpose of its first three columns, obtain in B a 3x3 matrix, and in CIR a 3x1 vector for the equations determining the normal direction of best-fit plane.

Call subroutine INV to solve equations in B and CIR.

Subroutine INV puts solution in CIR and determinant of B in DET.

If DET = 0?

Yes

Assuming that L > 1, generate CIR vector as cross product of vectors that join node ICN to first and second neighboring nodes.

By calling subroutine UNIT with Q > 0, generate a unit vector, in direction CIR, in CIR.

If INP < 2?

Yes

No

Print out overall coordinates of node ICN, and direction cosines of best-fit plane.

Return

Fig. VI-49. Flowchart of subroutine BEST (Link 4)
### SUBROUTINE BOFI

1. Set IS6 (number of elements of classes 5 and 6) and IH (number of elements of class 4) zero for node ICN. Also set IC = 4, IMS = 2.

2. Compute IS6 and IH values of node ICN by examining LM elements meeting at ICN.

3. Compute number of elements at node ICN that are not of classes 4, 5, and 6 into IRES.

4. Set ILIM 1, 2, or 3 if all LM elements are of classes 5 or 6, or of classes 1, 2, 3, or 7, or of class 4, respectively. Otherwise branch to block 90.

5. Set NB = 0.

6. Loop on elements (I) meeting at node ICN, from 1 to LM.

7. Obtain number of vertices of LM element of node ICN into JS1.

8. Find sequence number of nodes in vertices of LM element in J. If node ICN is not one of the vertices, branch to 19.

9. Yes: IMS < 2?
   - Yes: NB = 0
   - No: Loop on L satisfied?

10. Loop on I satisfied?

11. Loop on I satisfied?

12. Loop on K satisfied?

13. Augment NB by 1 and copy label of Ith neighbor of Jth vertex of the Lth element into Kth entry.


15. Loop on elements (I) by 3's (I is also count of subvectors of class 4).

16. Find sequence of vertices of element are simply sequential. First sequence numbers of vertices neighboring Jth vertex into JPI, JM1, and JS1.

17. Loop on elements of NSET matrix into IJ.

18. Loop on neighboring vertices (K) of Jth vertex.

19. Set ILIM < 3?

20. Septon L satisfied?

21. Three-dimensional mesh case. By setting K = 0, prepare to copy NSET vector into MSET vector by expanding.

22. Loop on elements of NSET vector (L) from 1 to 6 (a subvector has 6 elements).


24. Compute count of element of NSET matrix into IJ.


26. Loop on L satisfied?

27. Loop on I satisfied?

Fig. VI-50. Flowchart of subroutine BOFI (Link 4)
By setting KB as length of MSET vector and MB = 0, prepare to find out if node is on boundary, and if so, its neighbors on boundary.

Loop on elements (I) of MSET by 2s

Set I2 = 1 + 2

Node MSET(I) is not in NBAN list. Augment MB by 1 and copy MSET(I) as MBth element of NBAN

Node MSET(I) is in NBAN list. Augment MB by 1 and copy MSET(I) as MBth element of NBAN

Fig. Vi-50 (cont'd)
Loop on elements (J) of MSET from I to NB

Line or surface mesh case. Copy NSET vector into MSET and set MB = 3

Loop on elements (I) of MSET from 1 to NB

Set NODE = MSET(J), KLIM = 0

No

Yes

Fig. VI-50 (contd)
Fig. VI-50 (contd)
Necessary. Set $Q = -1$

$\text{SIGN} = \text{BIR} - \text{SIR} < 0$

It is not possible to streamline outer normal vector. Set INBON = 0 to ignore stress boundary conditions.

Normal is heading away from structure. Set $Q = 1$

Normal is heading towards structure; change sign if necessary. Set $Q = -1$

Call subroutine UNIT to compute average boundary generate unit normal vector. Set

$\text{SURFACE} = \text{BIR}$ as area tor on BIR of circle

By calling subroutine UNIT, produce desired unit normal vector on BIR.

Compute average distance between neighboring boundary nodes and node ICN, and assuming this is the radius, compute average boundary surface area on ARE as area of circle.

$\text{INP} < 2$

Yes

Print out ICN, AST, MB, NSAN-vector, MB, INSET vector, ARE, and BIR vector.

Return

Fig. VI-50 (contd)
SUBROUTINE DIMI (K)

100

Rewind tape ITAS

100

Set IERR = 0, ZGEM = 1, GEM, IDS = 2 * DCEG

110

Loop on elements (MM) of structure from 1 to IT

120

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

130

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

140

Clear COMMON area from 210 to 225

150

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

160

Call subroutine STRA to express in S the stiffness matrix in local coordinates

170

Obtain in VDE deflections of end points of Mth element in overall coordinates

180

By calling subroutine TRAN, express VDE in local coordinates

190

Obtain in DUM vector the product of S matrix and VDE vector as elastic forces at end

200

By calling subroutine TRAN, express fixed-end forces in local coordinates

210

Combine elastic-end forces with fixed-end forces to obtain final end forces in PV matrix in local coordinate system

220

Tape error. Set IERR = 1, K = M, 14(220) = MM

230

Print out element label, end-point labels, element type, and end forces in overall coordinates at end points

240

Loop on MM satisfied?

250

Tape error. Set IERR = 1, K = M, 14(250) = MM

260

No

270

Yes

280

Loop on elements

290

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

300

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

310

Clear COMMON area from 210 to 225

320

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

330

Call subroutine STRA to express in S the stiffness matrix in local coordinates

340

Obtain in VDE deflections of end points of Mth element in overall coordinates

350

By calling subroutine TRAN, express VDE in local coordinates

360

Obtain in DUM vector the product of S matrix and VDE vector as elastic forces at end

370

By calling subroutine TRAN, express fixed-end forces in local coordinates

380

Combine elastic-end forces with fixed-end forces to obtain final end forces in PV matrix in local coordinate system

390

Tape error. Set IERR = 1, K = M, 14(390) = MM

400

Print out element label, end-point labels, element type, and end forces in overall coordinates at end points

410

Loop on elements

420

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

430

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

440

Clear COMMON area from 210 to 225

450

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

460

Call subroutine STRA to express in S the stiffness matrix in local coordinates

470

Obtain in VDE deflections of end points of Mth element in overall coordinates

480

By calling subroutine TRAN, express VDE in local coordinates

490

Obtain in DUM vector the product of S matrix and VDE vector as elastic forces at end

500

By calling subroutine TRAN, express fixed-end forces in local coordinates

510

Combine elastic-end forces with fixed-end forces to obtain final end forces in PV matrix in local coordinate system

520

Tape error. Set IERR = 1, K = M, 14(520) = MM

530

Print out element label, end-point labels, element type, and end forces in overall coordinates at end points

540

Loop on elements

550

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

560

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

570

Clear COMMON area from 210 to 225

580

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

590

Call subroutine STRA to express in S the stiffness matrix in local coordinates

600

Obtain in VDE deflections of end points of Mth element in overall coordinates

610

By calling subroutine TRAN, express VDE in local coordinates

620

Obtain in DUM vector the product of S matrix and VDE vector as elastic forces at end

630

By calling subroutine TRAN, express fixed-end forces in local coordinates

640

Combine elastic-end forces with fixed-end forces to obtain final end forces in PV matrix in local coordinate system

650

Tape error. Set IERR = 1, K = M, 14(650) = MM

660

Print out element label, end-point labels, element type, and end forces in overall coordinates at end points

670

Loop on elements

680

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

690

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

700

Clear COMMON area from 210 to 225

710

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

720

Call subroutine STRA to express in S the stiffness matrix in local coordinates

730

Obtain in VDE deflections of end points of Mth element in overall coordinates

740

By calling subroutine TRAN, express VDE in local coordinates

750

Obtain in DUM vector the product of S matrix and VDE vector as elastic forces at end

760

By calling subroutine TRAN, express fixed-end forces in local coordinates

770

Combine elastic-end forces with fixed-end forces to obtain final end forces in PV matrix in local coordinate system

780

Tape error. Set IERR = 1, K = M, 14(780) = MM

790

Print out element label, end-point labels, element type, and end forces in overall coordinates at end points

800

Loop on elements

810

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

820

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

830

Clear COMMON area from 210 to 225

840

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

850

Call subroutine STRA to express in S the stiffness matrix in local coordinates

860

Obtain in VDE deflections of end points of Mth element in overall coordinates

870

By calling subroutine TRAN, express VDE in local coordinates

880

Obtain in DUM vector the product of S matrix and VDE vector as elastic forces at end

890

By calling subroutine TRAN, express fixed-end forces in local coordinates

900

Combine elastic-end forces with fixed-end forces to obtain final end forces in PV matrix in local coordinate system

910

Tape error. Set IERR = 1, K = M, 14(910) = MM

920

Print out element label, end-point labels, element type, and end forces in overall coordinates at end points

930

Loop on elements

940

Read Mth record from tape ITAS for M, ITTI, ITTM, NAV, IDS, IDS2, N, S, P1, and P2

950

Call subroutine CODI to obtain DIR matrix where rows are direction cosine of local axes

960

Clear COMMON area from 210 to 225

970

Obtain transpose of DIR matrix on DIR, for transformations from overall to local

980

Call subroutine STRA to express in S the stiffness matrix in local coordinates

990

Obtain in VDE deflections of end points of Mth element in overall coordinates

1000

By calling subroutine TRAN, express VDE in local coordinates

1010

Obtain XD, YD, ZD, and X, Y, Z vectors of Mth element

1020

Find element type IELT of Mth element

1030

Obtain XD, YD, ZD, and X, Y, Z vectors of Mth element from COMMON

Fig. VI-51. Flowchart of subroutine DIMI (Link 4)
SUBROUTINE DINA

1. Call subroutine AGEL to generate DIN matrix and ANGLE

2. General shell case. Call subroutine QUAD to obtain DIN matrix and ANGLE by fitting a quadratic surface on node set

3. Shell of revolution case. Call subroutine REVO to generate DIN matrix from a fourth-order polynomial curve fitted for meridian

4. Call subroutine AGEL to generate DIN matrix and ANGLE

5. INP < 2?

6. Print out DIN matrix (columns of which are direction cosines of local axes) and ANGLE

7. Return

Fig. VI-52. Flowchart of subroutine DINA (Link 4)
SUBROUTINE EPAN

Rearranges NB elements of NSET vector (labels of nodal set) in increasing order

10*
Set KK = ICN, MIN = 0

Loop on elements (J) of NSET vector from I to NB

30 K = NSET(J) - KK

10**

K < Q = Q = 0 ?

0
Set K = -K

101

Set NNN = NN, MIN = 0

Loop on elements (J) of MELE from I to NNN

250 NELE(J) = MELE(J)

255 Loop on L satisfied?

254 This is a new element. Augment NNN by 1 and copy MELE(J) into NELE(NNN)

255* Loop on J satisfied?

254* Forward tape ITAS K records

21

Read record for element set MELE of KKth node (llth node in node set of node ICN)

24* Fig. VI-53. Flowchart of subroutine EPAN (Link 4)
Start to reposition tape ITAS by setting K = ICN - KK

K < 0, N > 0, > 0 ?

< 0

Set K = -K

Backspace tape ITAS K records

Forward tape ITAS K records

Loop on elements (I) of NELE from 1 to NNN

Set in NN the element type and in K the material type of Ith element in NELE

If K = IMET, and NN is 11, 12, 13, or 14, first number of vertices in IMS; otherwise branch to 3

Set pointer of node labels of vertices of Ith element of NELE into JIII

Loop on vertices (J) of Ith element of NELE from 1 to IMS

Store label of Jth vertex of Ith element of NELE into K

Loop on entries (L) of NSET from 1 to NB

NSET(L) = K ?

Yes

K is a new node. Augment N8 by 1, and copy K into NSET(NB)

Loop on elements (I) of NSET(NB)

Loop on J satisfied ?

Yes

No

Yes

No

No

Yes

JPL TECHNICAL REPORT 32-1240

91
SUBROUTINE FINDQ(K, Q)

Compute pointer of deflections of node K into IDEF

Loop on degrees of freedom (i) of node K from 1 to IDEG

Find COMMON count of ith component of deflection vector of node K

Copy ith component of deflection vector of node K into Q(I)

Loop on I satisfied?

Return

SUBROUTINE GENE

Loop on elements (I) meeting at node ICN, from 1 to LM

Compute element label M of ith element in the element set listed in first column of NEL matrix

Obtain type numbers of descriptive information and node labels of Mth element by calling subroutine TOPO

Obtain number of vertices of Mth element in IAMS

Find class of Mth element and store into IC

Set LMT = LMT - 1 and print message that there are more than four material groups at LCL = 0

Copy X and Y coordinates of node CN into Q(1) and Q(2), respectively

Loop on I satisfied?

SUBROUTINE FINDX(K, Q)

Compute COMMON counts of X, Y, Z coordinates of ith node into IXXI, IYYI, IZZI

Copy X and Y coordinates of node K into Q(1) and Q(2), respectively

Compute Z coordinate of node K into Q(3)

Return

Fig. VI-54. Flowchart of subroutine FINDQ (Link 4)

Fig. VI-55. Flowchart of subroutine FINDX (Link 4)

Fig. VI-56. Flowchart of subroutine GENE (Link 4)
SUBROUTINE INER(CIR)

Make the three components of CIR vector zero

Loop on nodes (I) listed in NSET vector from 1 to NB

Find label of Ith node in NSET vector and store it into K

Obtain coordinates of node K in vector XF by calling subroutine FINDX

Augment CIR vector by vector indicated by line segment from node ICN to node K

No

Loop on I satisfied?

Yes

Return

SUBROUTINE INLZ

Set indicators IROT = 0 and BST = blank to imply that local axes are parallel to overall axes

Set DIN and W matrices as unit matrices of order 3 to imply that local axes are parallel to overall axes

Compute average TE, DT, and DG for ICh class of IMth material group at node ICN

Compute ICOL, BRG, and ERG values for ICh class of IMth material group at node ICN

Set ANGLE = 0 and ICON = IERR = 0

Set BAS vector as a unit vector heading from first vertex towards second vertex of lowest labelled element in ICh class of IMth material group at node ICN

Yes

Inp < 2?

No

Print out average TE, DT, ABS, and BAS vector

Return

Fig. VI-57. Flowchart of subroutine INER (Link 4)

Fig. VI-58. Flowchart of subroutine INLZ (Link 4)
SUBROUTINE INV (A, N, B, M, DETERM)

1. Set DETERM = 1.
2. Clear N elements in PIVAT vector.
4. Loop on rows (I) of coefficient matrix A from 1 to N:
   a. Set AMAX = 0.
   b. Loop on columns (J) of coefficient matrix A from 1 to N:
      i. If (Jth row is interchanged before)
      j. Loop on columns (K) of coefficient matrix A from 1 to N:
         k. If (ith column is interchanged before)
            l. Error
   m. Loop on rows (I) of coefficient matrix A from 1 to N:
      n. If (AMAX > BB) then
         o. Indicate that column ICOLUM is interchanged by augmenting PIVOT (ICOLUM) by 1.
      p. Start interchanging rows to put pivot on diagonal by setting DETERM = -DETERM.
      q. Interchange ICOLUMth row with IROWth row.
      r. Interchange ICOLUMth row with IROWth row in right-hand side(s).

Fig. VI-59. Flowchart of subroutine INV (Link 4)
Subtract $T$ multiple of $ICOLUMN$th row from $L1$th row in right-hand side(s).

Loop on $L1$ satisfied?

Loop on $I$ satisfied?

Loop on columns ($I$) of coefficient matrix from 1 to $N$

Set $L = N + 1 - 1$

Column interchange necessary?

Find column numbers of columns to be interchanged and store into $JROW$ and $JCOLUMN$ from index matrix

Interchange $JROW$th column with $JCOLUMN$th column

Loop on $I$ satisfied?

Error. Set $DETERM = 0.$

Return

Fig. VI-59 (contd)
SUBROUTINE LEST

Set ICONN = ICON, IERR = 0, JMAX = JMAX + JMR

Loop on rows (I) of B matrix from 1 to JMAX

Set (I, J) = 0

Loop on columns (J) of B matrix from 1 to JMM

Obtain symmetric element from B(J, I) = B(I, J)

Loop on I satisfied ?

Yes

Obtain symmetric element from 1 to JMM

Loop on J satisfied ?

Yes

Set B(I, J) = 0

Fig. VI-60. Flowchart of subroutine LEST (Link 4)
SUBROUTINE MDIN

Set indicators IMOT = 1, 
BST = 2H** to show that local coordinates are different than overall.

Yes ICAS = 4

Find overall coordinate axis that makes largest angle with normal vector BIR

By calling subroutine VECT, set ETA local axis as cross product of BIR and unit vector of Y-axis

Rotate local axes KSI, ETA, and ZTA such that KSI axis becomes coincident with outer normal BIR vector

No

Obtain a unit vector in ETA, using the vector in ETA. Obtain ZTA as (BIR)x(ETA) and KSI as (ETA)x(ZTA) so that KSI is coincident with BIR

Obtain ZTA as (BIR)x(ETA) and KSI as (ETA)x(ZTA) so that KSI is coincident with BIR

Fig. VI-61. Flowchart of subroutine MDIN (Link 4)
SUBROUTINE META

10

Clear material matrix DD

100

Set thermal constants AL1, AL2, and AL3 to zero

200

IF TYPE = 0, 1, or 2

300

= 2

= 1

No

= 0

60

Copy thermal constants into ADL, AL2, AL3, and upper material matrix into DD from COMMON

200

ISTR = 1

Yes

No

No

ICAS = 3 or 4

120

Set K = 3 and compute E1 and E2 for general solid case

120

Set K = 2 and compute E1 and E2 for plane stress case

300

ISTR = 1

Yes

No

Yes

Assume DD(1,3) and DD(0,3) to be identical with DD(1,2)

Assume DD(0,3) and DD(1,3) to be identical with DD(1,2)

60

Fig. VI-62. Flowchart of subroutine META (Link 4)
Loop on rows (I) of DD material matrix from 1 to K

Loop on columns (J) of DD material matrix from 1 to K

Set DD(I, J) = E1, DD(I+1, J-1) = G

Set DD(J, I) = E2

Interchange third and fourth rows of DD

Interchange third and fourth columns of DD. The matrix is now arranged in 1, 2, 12, 3, 13, 23 order

Print out material matrix of material type IMET in local coordinates in 1, 2, 12, 3, 13, 23 order

Generate symmetric lower part of material matrix DD

Fig. VI-62 (contd)
SUBROUTINE QUAD

Set K = 0 and IE as column number of last element in ICn row of 14th sheet of MAC matrix.

Loop on entries (i) of ICn row of 14th sheet of MAC matrix from 2 to IE.

Find sequence number (IEL) in NBL and number of vertices (IM) of finite element associated with ith entry.

Increment K by 1 and copy label of jth vertex into NSET(K).

Loop on vertices (j) from 1 to IM.

Set NODE = MSET(I), find sequence number (IELT) augment NB by 1, and copy NODE into NSET(NB).

Loop on 1 satisfied?

Yes

Set NB = 0.

Obtain first approximation of middle surface normal in ZD or (1-2)x(1-2) plane associated with MAC(IM, IC, 2).

Fig. VI-63. Flowchart of subroutine QUAD (Link 4)
SUBROUTINE REVO

Set necessary constants

Find number of elements of class IC and material IM, meeting at node ICN, into IE

IF IE = 1

Place first and second node labels of single element at first two elements into NSET array as second, third, and fourth entries

Obtain Jth entry in Ith row of B matrix (coefficient matrix of unknown factors of fourth-degree polynomial equation)

Compute power corresponding to J into L

Obtain right-hand side of 1th equation in first column of C

Loop on first two elements into NSET array as second, third, and fourth entries

Obtain label of Ith node into K

Obtain crude direction of first local axis XII as vector joining node IPONE to node IPONE in vector RED

Set IPONN = NB, UPP = IFONN + 1

Print out NSET vector to list labels of nodes to be used for polynomial curve

Obtain 4th and 5th word vectors of element descriptions, complete NSET vector to a vector of order 5 to contain labels of nodes for polynomial curve. The order of NSET is in NB and NSET is always left justified, and contains labels of consecutive nodes pertaining to elements of class IC and material IM

Loop on nodes (I) included for use in basis fit curve

Compute power corresponding to J into L

Obtain right-hand side of 1th equation in first column of C

Print out NSET vector to list labels of nodes to be used for polynomial curve

Yes

No

Yes

No

Yes

No

Yes

No

Fig. VI-64. Flowchart of subroutine REVO (Link 4)
Set unit vector of second local axis ETA as reverse of that of overall Z-axis.

Set $Q = 1$.

Obtain unknown coefficients of polynomial by solving the set in B and C. Solution is stored in C, and determinant of B is in DET.

Obtain unit normal in RED copy RED into XII; set ZTA as unit vector in (XII)$ \times$ (ETA) direction.

Obtain unit vector in RED. Loop on INPONN complement of integer powers (I) from 1 to IFF.

Compute power corresponding to I into K, and set $C_I = K$.

Compute contribution of Kth power term into first derivative of $Y(X)$ evaluated at node ICN and add to ZTA(1).

Loop on I satisfied?

Set ZTA(2) = -1, ZTA(3) = 0.

Obtain unit normal in ZTA heading to proper direction; then obtain XII as unit vector in (ETA)$ \times$ (ZTA) direction.

Obtain XII as (ETA) $\times$ (ZTA).

Yes

$Q(\text{RED}) (XII) > 0$?

No

Set $Q = -1$, since normal in ZTA is heading in wrong direction.

Obtain unit normal in ZTA heading to proper direction; then obtain XII as unit vector in (ETA)$ \times$ (ZTA) direction.

Yes

No

Fig. VI-64 (contd)
SUBROUTINE ROTA

19 Material Isotropic ?
   Yes
   ICAS = 3 ?
   Yes
   Copy DIN matrix into W matrix, which is direction cosine matrix of new material axes in old ones
   No
   ICAS = 4 ?
   Yes
   Yes
   ICAS = 5 ?
   No
11 Compute angle between first material AX(BAS) and first local axis (XI) into AI
   No
   Yes
   ICAS = 6 ?
   No
   10 Obtain direction cosines of material axes in overall system as columns of V matrix as (ZTA x BAS), (ZTA) x (BAS), (ZTA)
   11 Express V matrix in local coordinates into W matrix, which has some meaning as in block 11
   No
   Yes
   Angle AG less than 1 deg ?
   101
   Yes
   Observe direction cosines of material axes in overall system as columns of V matrix as (ZTA x (BAS)), (ZTA) x (BAS), AND (ZTA)
   No
   15 Return

200

16 Compute material transformation matrix into R by using matrix W
20 Obtain in T the matrix product of (R) (DD)
21 Obtain in DD the matrix product of (T)(R), which is material matrix in new axes
19 INP < 2 ?
   No
   Yes
20 Print out matrix R
200

Return

Fig. VI-65. Flowchart of subroutine ROTA (Link 4)
SUBROUTINE SAME

Set BCD blank in TEST

110

BJST = TEST ?

Yes

ICAS > 4 ?

Yes

ICAS = 2 ?

Yes

Obtain symmetric lower half of stress tensor in DMM

Obtain in DMM the matrix product of (DMM) (DIN)

Obtain in DMM the matrix product of (DIN)T (DMMM), which is stress tensor in overall coordinates

ICAS = 2 ?

Yes

Copy bending moments from DMM into SR (reverse operation of block 112)

ICAS < 4 ?

Yes

110

Return

110

No

112

No

111

No

ICAS = 2 ?

Yes

114

Obtain in DMM the matrix product of (DMM) (DIN)

Obtain in DMM the matrix product of (DIN)T (DMMM), which is moment of stress tensor in overall coordinates

112

No

113

Stresses are in RED vector in order of 1, 2, 3, 2, 3, 3. Generate stress tensor in 3x3 DMM matrix

111

No

110

No

100

110

Yes

200

Obtain symmetric lower half of stress tensor in DMM

400

Obtain in DMM the matrix product of (DMM) (DIN)

100

100

Return

Fig. VI-66. Flowchart of subroutine SAME (Link 4)
Fig. VI-67. Flowchart of function SCAL (Link 4)

Fig. VI-68. Flowchart of subroutine SETA (Link 4)
Se $C = 0$ 

$CC = 0$. 

py 45

Set $CC = 0$. 

DT = 0 

Ye 

No 

ITYPE = 0? 

Yes 

No 

Call subroutine TEMP to compute into $CC$ thermal strain per unit temperature; then set $CC = CC \times DT$. 

59* 

Compute and store properly left-hand side of $ICON$th equation. 

ITYPE = 0? 

Yes 

No 

Call subroutine TEMP to compute into $CC$ thermal rotation per unit gradient; then set $CC = CC \times DG$. 

59* 

Compute component of relative rotation vector on nodal line. 

Subtract nodal line component of relative rotation vector from itself and obtain $DC$ component in direction normal to (ZTA-nodal line) plane. Set $DC = DC/CL$. 

Obtain left-hand side of $ICON$th equation. 

Fig. VI-68 (contd)
SUBROUTINE STRS

1040 Set CM = TE

Set CM for bending moments as CM = CM*TE*TE/12.

Interchange third and fourth elements in SR vector so that stresses are in 1, 2, 12, 3, 13, and 23 order.

Return

Fig. VI-69. Flowchart of subroutine STRS (Link 4)
SUBROUTINE TEMP(CC)

Express unit vector of nodal line (ICN-K) in original material axes into SIR vector

Using SIR vector, compute elongation of unit distance on SIR direction due to 1-deg temperature rise

Return

Fig. VI-70. Flowchart of subroutine TEMP (Link 4)

SUBROUTINE VECT(A, B, C)

SUBROUTINE UNIT(CIR, Q)

Obtain in vector A the cross product (B) x (C)

Compute length square of vector in CIR into R

Set Q = R

Change direction of CIR vector

Set Q = V/R and scale down CIR by Q to obtain unit vector

Return

Fig. VI-72. Flowchart of subroutine VECT (Link 4)

Fig. VI-71. Flowchart of subroutine UNIT (Link 4)
VII. Source Program Listings

This section contains the source program listings of ELAS/Level 3. The listing of each program element is treated separately, and given a table number. The listings are arranged alphabetically by the subroutine names, under the main program of each link. The meanings of the variables used in the source program may be obtained from Tables III-2, III-3, and III-4 of Vol. II (basic). The organization of COMMON for each link is shown in Fig. III-1 of Vol. II (basic).
Table VII-1 (contd)

Table VII-1: Table VII-1 (contd)

<table>
<thead>
<tr>
<th>Column 1</th>
<th>Column 2</th>
<th>Column 3</th>
<th>Column 4</th>
<th>Column 5</th>
<th>Column 6</th>
<th>Column 7</th>
<th>Column 8</th>
<th>Column 9</th>
<th>Column 10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table VII-1 (contd)**

**Table VII-1 (contd)**
Table VII-2 (contd)

Table VII-3. Source program listing of subroutine BUNG (Link 1)
<table>
<thead>
<tr>
<th>Table VII-8. Source program listing of subroutine MESS (Link 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># LABEL</strong></td>
</tr>
<tr>
<td><strong>C SUBROUTINE MESS</strong></td>
</tr>
<tr>
<td><strong>C DUMMY SUBROUTINE FOR MESS CONVERSION</strong></td>
</tr>
<tr>
<td><strong>RETURN</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table VII-9. Source program listing of subroutine MEST (Link 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># LABEL</strong></td>
</tr>
<tr>
<td><strong>C SUBROUTINE MEST</strong></td>
</tr>
<tr>
<td><strong>C DUMMY SUBROUTINE FOR MEST CONVERSION</strong></td>
</tr>
<tr>
<td><strong>RETURN</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table VII-10. Source program listing of subroutine OUTPT (Link 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong># LABEL</strong></td>
</tr>
<tr>
<td><strong>C SUBROUTINE OUTPT</strong></td>
</tr>
<tr>
<td><strong>C DUMMY SUBROUTINE FOR OUTPT CONVERSION</strong></td>
</tr>
<tr>
<td><strong>RETURN</strong></td>
</tr>
</tbody>
</table>
SUBROUTINE SRAT

* IF I = 1 THEN 220
* ELSE IF I = 2 THEN 300
* ELSE IF I = 3 THEN 380
* ELSE IF I = 4 THEN 460
* ELSE IF I = 5 THEN 540

CONTINUE

DO 10 L = 1, N
* IF L = N THEN 200
* ELSE IF L = 1 THEN 200
* ELSE IF L = 2 THEN 200
* ELSE IF L = 3 THEN 200
* ELSE IF L = 4 THEN 200
* ELSE IF L = 5 THEN 200

CONTINUE

DO 100 I = 1, N
* IF I = 1 THEN 200
* ELSE IF I = 2 THEN 200
* ELSE IF I = 3 THEN 200
* ELSE IF I = 4 THEN 200
* ELSE IF I = 5 THEN 200

CONTINUE

END

1/1

INTERFACE

-CONNECTIVITY

100 CONTINUE

781 CONTINUE

799 DO 100 J = 1, M
* IF J = 1 THEN 200
* ELSE IF J = 2 THEN 200
* ELSE IF J = 3 THEN 200
* ELSE IF J = 4 THEN 200
* ELSE IF J = 5 THEN 200

CONTINUE

END
Table VII-13. Source program listing of subroutine TICK (link 1)

Table VII-12. Source program listing of subroutine TABL (link 1)

Table VII-14. Source program listing of subroutine TOPO (link 1)
Table VII-14 (contd)

<table>
<thead>
<tr>
<th>đời disple</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2113</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2114</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2115</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2116</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2117</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2118</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2119</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2120</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table VII-15. Source program listing of main program of Link 2 (generation link)

<table>
<thead>
<tr>
<th>đời disple</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>2113</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2114</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2115</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2116</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2117</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2118</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2119</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>2120</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

JPL TECHNICAL REPORT 32-1240

121
<table>
<thead>
<tr>
<th>Table VII-15 (contd)</th>
</tr>
</thead>
</table>

```
600 IF (INS-11.11) .GE. 1.11)
601 GO TO 600
602 CONTINUE
603 IF (INS-11.11) .GE. 1.11)
604 CONTINUE
605 IF (INS-11.11) .GE. 1.11)
606 CONTINUE
607 IF (INS-11.11) .GE. 1.11)
608 CONTINUE
609 CONTINUE

610 IF (INS-11.11) .GE. 1.11)
611 GO TO 610
612 CONTINUE
613 IF (INS-11.11) .GE. 1.11)
614 CONTINUE
615 IF (INS-11.11) .GE. 1.11)
616 CONTINUE
617 IF (INS-11.11) .GE. 1.11)
618 CONTINUE
619 CONTINUE

620 IF (INS-11.11) .GE. 1.11)
621 GO TO 620
622 CONTINUE
623 IF (INS-11.11) .GE. 1.11)
624 CONTINUE
625 IF (INS-11.11) .GE. 1.11)
626 CONTINUE
627 IF (INS-11.11) .GE. 1.11)
628 CONTINUE
629 CONTINUE

630 IF (INS-11.11) .GE. 1.11)
631 GO TO 630
632 CONTINUE
633 IF (INS-11.11) .GE. 1.11)
634 CONTINUE
635 IF (INS-11.11) .GE. 1.11)
636 CONTINUE
637 IF (INS-11.11) .GE. 1.11)
638 CONTINUE
639 CONTINUE

640 IF (INS-11.11) .GE. 1.11)
641 GO TO 640
642 CONTINUE
643 IF (INS-11.11) .GE. 1.11)
644 CONTINUE
645 IF (INS-11.11) .GE. 1.11)
646 CONTINUE
647 IF (INS-11.11) .GE. 1.11)
648 CONTINUE
649 CONTINUE

650 IF (INS-11.11) .GE. 1.11)
651 GO TO 650
652 CONTINUE
653 IF (INS-11.11) .GE. 1.11)
654 CONTINUE
655 IF (INS-11.11) .GE. 1.11)
656 CONTINUE
657 IF (INS-11.11) .GE. 1.11)
658 CONTINUE
659 CONTINUE

660 IF (INS-11.11) .GE. 1.11)
661 GO TO 660
662 CONTINUE
663 IF (INS-11.11) .GE. 1.11)
664 CONTINUE
665 IF (INS-11.11) .GE. 1.11)
666 CONTINUE
667 IF (INS-11.11) .GE. 1.11)
668 CONTINUE
669 CONTINUE

670 IF (INS-11.11) .GE. 1.11)
671 GO TO 670
672 CONTINUE
673 IF (INS-11.11) .GE. 1.11)
674 CONTINUE
675 IF (INS-11.11) .GE. 1.11)
676 CONTINUE
677 IF (INS-11.11) .GE. 1.11)
678 CONTINUE
679 CONTINUE

680 IF (INS-11.11) .GE. 1.11)
681 GO TO 680
682 CONTINUE
683 IF (INS-11.11) .GE. 1.11)
684 CONTINUE
685 IF (INS-11.11) .GE. 1.11)
686 CONTINUE
687 IF (INS-11.11) .GE. 1.11)
688 CONTINUE
689 CONTINUE

690 IF (INS-11.11) .GE. 1.11)
691 GO TO 690
692 CONTINUE
693 IF (INS-11.11) .GE. 1.11)
694 CONTINUE
695 IF (INS-11.11) .GE. 1.11)
696 CONTINUE
697 IF (INS-11.11) .GE. 1.11)
698 CONTINUE
699 CONTINUE

700 IF (INS-11.11) .GE. 1.11)
701 GO TO 700
702 CONTINUE
703 IF (INS-11.11) .GE. 1.11)
704 CONTINUE
705 IF (INS-11.11) .GE. 1.11)
706 CONTINUE
707 IF (INS-11.11) .GE. 1.11)
708 CONTINUE
709 CONTINUE
```
```
Table VII-20. Source program listing of subroutine CORT (Link 2)

Table VII-21. Source program listing of subroutine CUTE (Link 2)
Table VII-23. Source Program listing of subroutine DMN (Link 2)

Table VII-24. Source program listing of subroutine EDD (Link 2)
Table VII-30. Source program listing of subroutine S04 (Link 2)

Table VII-31. Source program listing of subroutine S05 (Link 2)
Table VII-32. Source program listing of subroutine S07 (Link 2)

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CALL LABREL(301,302)</td>
</tr>
<tr>
<td>B</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
<tr>
<td>C</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
<tr>
<td>D</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
<tr>
<td>E</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
</tbody>
</table>

Table VII-33. Source program listing of subroutine S09 (Link 2)

<table>
<thead>
<tr>
<th>Label</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>CALL LABREL(301,302)</td>
</tr>
<tr>
<td>B</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
<tr>
<td>C</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
<tr>
<td>D</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
<tr>
<td>E</td>
<td>CALL TRM IE22IA.Q2,3</td>
</tr>
</tbody>
</table>

Note: The table contains source program listings for subroutines S07 and S09, which are part of the JPL TECHNICAL REPORT 32-1240.
### Table VII-33 (contd)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H2</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H3</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H4</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H5</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H6</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H7</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H8</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H9</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H10</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H11</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H12</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H13</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H14</td>
<td>(x_i, y_i, z_i)</td>
</tr>
<tr>
<td>H15</td>
<td>(x_i, y_i, z_i)</td>
</tr>
</tbody>
</table>

### Table VII-34. Source program listing of subroutine S11 (Link 2)

**Label:** CB201

**Generates for element type: 1 (stiffness and load matrices)**

<table>
<thead>
<tr>
<th>Source Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
</tr>
<tr>
<td>2.</td>
</tr>
<tr>
<td>3.</td>
</tr>
<tr>
<td>4.</td>
</tr>
<tr>
<td>5.</td>
</tr>
<tr>
<td>6.</td>
</tr>
<tr>
<td>7.</td>
</tr>
<tr>
<td>8.</td>
</tr>
<tr>
<td>9.</td>
</tr>
<tr>
<td>10.</td>
</tr>
</tbody>
</table>

### Table VII-35. Source program listing of subroutine S13 (Link 2)

**Label:** CB211

**Generates for element type: 3 (stress) and load matrices**

<table>
<thead>
<tr>
<th>Source Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
</tr>
<tr>
<td>2.</td>
</tr>
<tr>
<td>3.</td>
</tr>
<tr>
<td>4.</td>
</tr>
<tr>
<td>5.</td>
</tr>
<tr>
<td>6.</td>
</tr>
<tr>
<td>7.</td>
</tr>
<tr>
<td>8.</td>
</tr>
<tr>
<td>9.</td>
</tr>
<tr>
<td>10.</td>
</tr>
</tbody>
</table>
Table VII-44. Source program listing of subroutine TRIM (Link 2)

<table>
<thead>
<tr>
<th>LABEL</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRIM</td>
<td></td>
</tr>
<tr>
<td>CE2TRI</td>
<td></td>
</tr>
<tr>
<td>F2TRIO01</td>
<td></td>
</tr>
<tr>
<td>CEZTRM</td>
<td></td>
</tr>
<tr>
<td>N=-N</td>
<td></td>
</tr>
<tr>
<td>F2TRMOnS</td>
<td></td>
</tr>
<tr>
<td>(IA(38),IORD),IA(39),ACEL),IA(50),J1),IA(51),J2),F2TRIO13</td>
<td></td>
</tr>
<tr>
<td>41STR),(IA128),IELT)I,IA129),ITEM),(IA3 ),ITIlC),IA)31),IMET),R),AA(186I</td>
<td></td>
</tr>
<tr>
<td>4(AA(8R3),ALL</td>
<td></td>
</tr>
<tr>
<td>ICIZ),(IA</td>
<td></td>
</tr>
<tr>
<td>A),</td>
<td></td>
</tr>
<tr>
<td>CONTINUE F2TRM</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

DIMENSION A(3,3),R(4,4) .C()4,4) |  |
SUBROUTINE TRM |  |
END |  |

Table VII-45. Source program listing of subroutine TRM (Link 2)

<table>
<thead>
<tr>
<th>LABEL</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CTRIM</td>
<td></td>
</tr>
<tr>
<td>CE2TRI</td>
<td></td>
</tr>
<tr>
<td>F2TRIO01</td>
<td></td>
</tr>
<tr>
<td>CEZTRM</td>
<td></td>
</tr>
<tr>
<td>N=-N</td>
<td></td>
</tr>
<tr>
<td>F2TRMOnS</td>
<td></td>
</tr>
<tr>
<td>(IA(38),IORD),IA(39),ACEL),IA(50),J1),IA(51),J2),F2TRIO13</td>
<td></td>
</tr>
<tr>
<td>41STR),(IA128),IELT)I,IA129),ITEM),(IA3 ),ITIlC),IA)31),IMET),R),AA(186I</td>
<td></td>
</tr>
<tr>
<td>4(AA(8R3),ALL</td>
<td></td>
</tr>
<tr>
<td>ICIZ),(IA</td>
<td></td>
</tr>
<tr>
<td>A),</td>
<td></td>
</tr>
<tr>
<td>CONTINUE F2TRM</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
</tbody>
</table>

DIMENSION A(3,3),R(4,4) .C()4,4) |  |
SUBROUTINE TRM |  |
END |  |

JPL TECHNICAL REPORT 32-1240
Table VII-46. Source program listing of main program of Link 3 (deflection link)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BEGIN PROGRAM MAIN</td>
</tr>
<tr>
<td>2</td>
<td>USE MEASURES INITIAL COORDINATES</td>
</tr>
<tr>
<td>5</td>
<td>EQUIVALENCE (M, N, K, L, P, Q, R, S, T, U, V, W, X, Y, Z)</td>
</tr>
<tr>
<td>6</td>
<td>T=T+1</td>
</tr>
<tr>
<td>7</td>
<td>IF (I+1) 302, 303, 304</td>
</tr>
<tr>
<td>8</td>
<td>T=T+1</td>
</tr>
<tr>
<td>9</td>
<td>IF (I+1) 302, 303, 304</td>
</tr>
<tr>
<td>11</td>
<td>T=T+1</td>
</tr>
<tr>
<td>12</td>
<td>IF (I+1) 302, 303, 304</td>
</tr>
<tr>
<td>13</td>
<td>WRITE OUTPUT TAPE 5, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13</td>
</tr>
<tr>
<td>14</td>
<td>END</td>
</tr>
</tbody>
</table>

Note: The listing continues with more detailed source code information.
Table VII-47. Source program listing of subroutine ELST (Link 3)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE ELST</td>
</tr>
<tr>
<td>2</td>
<td>(CONTINUED)</td>
</tr>
<tr>
<td>3</td>
<td>C WRITES ON TAP 8 (TAB CLEMENT SET INFORMATION)</td>
</tr>
<tr>
<td>4</td>
<td>DIMENSION (IA(1),AA(1)),AA(1),IA(1),IA(2),IA(3),IA(4)</td>
</tr>
<tr>
<td>5</td>
<td>1(IA(1)),AA(1),IA(1),IA(2),IA(3),IA(4)</td>
</tr>
<tr>
<td>6</td>
<td>C WRT IN GEE</td>
</tr>
<tr>
<td>7</td>
<td>EQUIVALENCE (AA(1),AA(2),AA(3),AA(4),AA(5),AA(6))</td>
</tr>
<tr>
<td>8</td>
<td>EQUIVALENCE (AA(1),AA(2),AA(3),AA(4),AA(5),AA(6))</td>
</tr>
<tr>
<td>9</td>
<td>SUBPROGRAMS (AA(1),AA(2),AA(3),AA(4),AA(5),AA(6))</td>
</tr>
<tr>
<td>10</td>
<td>SUBPROGRAMS (AA(1),AA(2),AA(3),AA(4),AA(5),AA(6))</td>
</tr>
<tr>
<td>11</td>
<td>WRITE (AA(1),AA(2),AA(3),AA(4),AA(5),AA(6))</td>
</tr>
<tr>
<td>12</td>
<td>WRITE (AA(1),AA(2),AA(3),AA(4),AA(5),AA(6))</td>
</tr>
<tr>
<td>13</td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>

Table VII-48. Source program listing of subroutine PUNC (Link 3)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE PUNC</td>
</tr>
<tr>
<td>2</td>
<td>(CONTINUED)</td>
</tr>
<tr>
<td>3</td>
<td>C WRITE ON TAP 9</td>
</tr>
<tr>
<td>4</td>
<td>DIMENSION (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>5</td>
<td>1(IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>6</td>
<td>C WRITE ON TAP 9</td>
</tr>
<tr>
<td>7</td>
<td>EQUIVALENCE (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>8</td>
<td>EQUIVALENCE (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>9</td>
<td>SUBPROGRAMS (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>10</td>
<td>SUBPROGRAMS (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>11</td>
<td>WRITE (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>12</td>
<td>WRITE (IA(1),IA(2),IA(3),IA(4),IA(5),IA(6))</td>
</tr>
<tr>
<td>13</td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>

Table VII-49. Source program listing of subroutine RESI (Link 3)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SUBROUTINE RESI</td>
</tr>
<tr>
<td>2</td>
<td>(CONTINUED)</td>
</tr>
<tr>
<td>3</td>
<td>C WRITE RESI ON TAP 10</td>
</tr>
<tr>
<td>4</td>
<td>DIMENSION (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>5</td>
<td>1(IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>6</td>
<td>C WRITE RESI ON TAP 10</td>
</tr>
<tr>
<td>7</td>
<td>EQUIVALENCE (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>8</td>
<td>EQUIVALENCE (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>9</td>
<td>SUBPROGRAMS (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>10</td>
<td>SUBPROGRAMS (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>11</td>
<td>WRITE (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>12</td>
<td>WRITE (IA(1),AA(1),AA(2),AA(3),AA(4),AA(5))</td>
</tr>
<tr>
<td>13</td>
<td>CONTINUE</td>
</tr>
</tbody>
</table>

JPL TECHNICAL REPORT 32-1240
Table VII-50. Source program listing of subroutine RESW (Link 3)

```
* LABEL
CP-NAME
SUBROUTINE RESW
1

WRITE 4,6.160,6.168,10,500,6,168,250,6,168,210

COMMON /AA/...

C EQUIVALENCE (AA(1),E(1)),(AA(2),E(2)),(AA(3),E(3)),(AA(4),E(4)),
...(AA(100),E(100))

END
TRA 2.4
STL
CAL
TRA
COUNT 25
FAP
```

Table VII-51. Source program listing of subroutine TICK (Link 3)

```
* LABEL
CP-NAME
SUBROUTINE TICK
1

WRITE 4,6.160,6.168,10,500,6,168,250,6,168,210

COMMON /AA/...

C EQUIVALENCE (AA(1),E(1)),(AA(2),E(2)),(AA(3),E(3)),(AA(4),E(4)),
...(AA(100),E(100))

END
```

138

JPL TECHNICAL REPORT 32.
Table VII-52. Source program listing of subroutine VELAS (Link 3)

```
* LABEL

C PROGRAM LISTING OF SUBROUTINE VELAS (LINK 3)

C SOLVES THE GOVERNING EQUATIONS BY VARIABLE BAND-WIDTH CHILESKI METHOD

DIMENSION A(IST+1,IST+1)
COMMON (A,IST)
EQUALEMENT (A,IST)
JERKE
I=IST+1
J=1
IF (I+N-1 GT 1000) GO TO 101
I=1
GO TO 100

100 CONTINUE

C FIND THE SMALLEST DIAGONAL ELEMENT.
101 I=1
I=IST+1
J=1
I=IST+1
J=1
GO TO 109
T6 IF (A(IJ)=0 THEN 10,11,12
T7 I=IST+1
CONTINUE
C SET ALLOCABLE MINIMUM ON DIAGONAL ELEMENTS.
C
C OBTAIN U1,1
T8 IF (I=1 THEN 11,13
T9 I=IST+1
CONTINUE
C OBTAIN THE REST OF FIRST ROW OF U.
110 E=0
I=IST+1
K=1
111 I=IST+1
K=1
GO TO 117
117 I=IST+1
K=1
GO TO 113
113 DO 131 I=1,N
131 DO 131 11,12
132 CONTINUE
C OBTAIN THE OTHER ROWS OF U SEQUENTIALLY.
133 CONTINUE
C
C RETURN SWEEP IS COMPLETED, START BACKWARD SWEEP, FIRST.

C CHECK COMPLETENESS.

C TURN THE REST OF U IN BACKWARD DIRECTION.

C CHECK COMPLETENESS.

C CHECK COMPLETENESS.

C REPORT
```

---

The source program listing of subroutine VELAS (Link 3) is shown above. The listing includes a series of statements that solve the governing equations using the variable band-width ChILESki method. It involves finding the smallest diagonal element, obtaining the first row, and then obtaining the rest of the rows sequentially. The program includes loops and conditional statements to handle the calculations efficiently.
Table VII-54. Source program listing of subroutine ABEQ (Link 4)

<table>
<thead>
<tr>
<th>Line</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>201</td>
<td>( \text{CONTINUE} )</td>
</tr>
<tr>
<td>202</td>
<td>( \text{GO TO 15} )</td>
</tr>
<tr>
<td>203</td>
<td>( \text{RETURN} )</td>
</tr>
<tr>
<td>204</td>
<td>( \text{STOP} )</td>
</tr>
</tbody>
</table>

Table VII-55. Source program listing of subroutine BEST (Link 4)

Table VII-56. Source program listing of subroutine AGEL (Link 4)

Table VII-57. Source program listing of subroutine BOF1 (Link 4)
Table VII-57 (contd)

Table VII-58. Source program listing of subroutine CAS4 (Link 4)
<table>
<thead>
<tr>
<th>Table VII-59: Source program listing of subroutine CODI (Link 4)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Table VII-60: Source program listing of subroutine DIMI (Link 4)</strong></td>
</tr>
</tbody>
</table>

**Table VII-59: Source program listing of subroutine CODI (Link 4)**

```
GO TO 1000
```

**Table VII-60: Source program listing of subroutine DIMI (Link 4)**

```
GO TO 1000
```
Table VII-60 (contd)

[Code listing for technical report, not transcribed here]

Table VII-61. Source program listing of subroutine DINA (Link 4)

<table>
<thead>
<tr>
<th>LABEL</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>TO GENERATE GAS VELOC (O2 PART AND ANGL)</td>
</tr>
<tr>
<td>C</td>
<td>OBTAINS LOCAL DIMENSIONS IA(II,AA(1I),1.)N(BlO21121),033(3,3),E22(3,3)</td>
</tr>
<tr>
<td>C</td>
<td>GENERATE FOR THE LOCAL COORDINATE AXES AT A NODE ON SHELLS</td>
</tr>
<tr>
<td>C</td>
<td>IF)1,AA8214).Nl8),(AA215,9'</td>
</tr>
<tr>
<td>C</td>
<td>(AA(84),AL2)(A(85),AL3),(AA(86).021)</td>
</tr>
<tr>
<td>C</td>
<td>(AA(207,IMFL)L</td>
</tr>
<tr>
<td>C</td>
<td>(AA201).ICN).IAA(202),LM),(AA( 203,AST4</td>
</tr>
</tbody>
</table>
Table VII-62. Source program listing of subroutine EPAN (Link 4)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>C ORDER THE NODES IN NSET AS INCREASING</td>
</tr>
</tbody>
</table>
Table VII-65. Source program listing of subroutine GENE (Link 4)

<table>
<thead>
<tr>
<th>Label</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LABEL</td>
<td>CEAGEN</td>
<td>SUBROUTINE GENE</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN GENE</td>
<td>GENERATES INCL AND MACR AT ID'S IF A NMIN</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN DIMENSION</td>
<td>THE INCL AND ID'S (1041X12)</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN NSET, KLM, NELC</td>
<td>(NSET, KLM, NELC)</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN ITM</td>
<td>ITM=13</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN IC=8</td>
<td>IC=8</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN IC=4</td>
<td>IC=4</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN IMS=2</td>
<td>IMS=2</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN LS</td>
<td>LS=3</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN DO 23</td>
<td>DO 23</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN KJ, LM</td>
<td>KJ, LM</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN DO 33</td>
<td>DO 33</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN J, L</td>
<td>J, L</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN DO 34</td>
<td>DO 34</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN JS, Z</td>
<td>JS, Z</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN ED6</td>
<td>ED6</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN SSBA</td>
<td>SSBA</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN IN</td>
<td>IN</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN 23, 233, 234</td>
<td>23, 233, 234</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN END</td>
<td>END</td>
</tr>
</tbody>
</table>

Table VII-66. Source program listing of subroutine INNER (Link 4)

<table>
<thead>
<tr>
<th>Label</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LABEL</td>
<td>CEAGEN</td>
<td>SUBROUTINE INNER</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN INNER</td>
<td>GENATES A VECTOR HEADING TOWARDS THE STRUCTURE AT A BOUNDARY NO.</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN DIMENSION</td>
<td>THE INCL AND ID'S (1041X12)</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN GENE</td>
<td>GENE</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN 1,233, 233, 233</td>
<td>1,233, 233, 233</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN ED6</td>
<td>ED6</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN SSBA</td>
<td>SSBA</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN IN</td>
<td>IN</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN 23, 233, 234</td>
<td>23, 233, 234</td>
</tr>
<tr>
<td>C</td>
<td>CEAGEN END</td>
<td>END</td>
</tr>
</tbody>
</table>

148 JPL TECHNICAL REPORT 32-1240
Table VII-47. Source program listing of subroutine INLZ (Link 4)

```
C INITIALIZE SCALARS, VECTORS AND MATRICES AT CE
C
C SET INDICATORS

DIMENSION IA(1), AA(1), (N,1), D21(21), D33(3,3), E22(3,3)

DO I = 1, 21
  DO J = 1, 21
    IA(I,J) = 0
  ENDDO
ENDDO

C EQUIVALENCE (NES(1)
C EQUIVALENCE (AA(1220), RI(1), AA(223), SIR, AA(226), IN)
C EQUIVALENCE (AA(200), IO(1), AA(201), ICN, AA(202), LM, AA(203), AST)
C EQUIVALENCE (AA(214), NB, AA(215), RR)

C CONTINUE

CONTINUE
```
Table VII-68. Source program listing of subroutine INV (Link 4)

Table VII-69. Source program listing of subroutine LEST (Link 4)
Table VII-70. Source program listing of subroutine MDIN (link 4)
Table VII-72. Source program listing of subroutine QUAD (Link 4)

<table>
<thead>
<tr>
<th>Column</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A ALTER</td>
</tr>
</tbody>
</table>
Table VII-73. Source program listing of subroutine REVO (Link 4)

<table>
<thead>
<tr>
<th>Line</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BEGINNING OF PROGRAM</td>
</tr>
<tr>
<td>2</td>
<td>FUNCTION OF POLYNOMIAL</td>
</tr>
<tr>
<td>3</td>
<td>DIMENSION</td>
</tr>
<tr>
<td>4</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>5</td>
<td>SUBROUTINE REVO</td>
</tr>
<tr>
<td>6</td>
<td>R optimized for shells in revolution at a given angle</td>
</tr>
<tr>
<td>7</td>
<td>DIMENSION</td>
</tr>
<tr>
<td>8</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>9</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>10</td>
<td>DIMENSION</td>
</tr>
<tr>
<td>11</td>
<td>SUBROUTINE REVO</td>
</tr>
<tr>
<td>12</td>
<td>R optimized for shells in revolution at a given angle</td>
</tr>
<tr>
<td>13</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>14</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>15</td>
<td>DIMENSION</td>
</tr>
<tr>
<td>16</td>
<td>SUBROUTINE REVO</td>
</tr>
<tr>
<td>17</td>
<td>R optimized for shells in revolution at a given angle</td>
</tr>
<tr>
<td>18</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>19</td>
<td>EQUIVALENCE</td>
</tr>
<tr>
<td>20</td>
<td>DIMENSION</td>
</tr>
<tr>
<td>21</td>
<td>SUBROUTINE REVO</td>
</tr>
<tr>
<td>22</td>
<td>R optimized for shells in revolution at a given angle</td>
</tr>
</tbody>
</table>

*NOTE:* The above text is a transcription of the source program listing of subroutine REVO (Link 4) from the JPL Technical Report 32-1240.
The angle between the axes of the new material axes and the old system.

Table VII/4: Source program listing of subroutine ROTA (Link 4)
C REARRANGE SR NOTING THE SHIFT IN COMUTTION
750 CONTINUE (F4SAM ,08 08)
31 CAS),(A A(204),F I)
2(AA(204),I ROT),(AA 205), 7,(AA(42),IFO),(AA(43),IPG)0,(AA
6(AA(8 6),Y),(AA()351)S),(AA(4
5(AA1
4(AAI83),AL1
2(IA(74),
911A),(IA(63),IDT),(IA(64),IDY),(AI651) ,TE)I(I A(1))
EI,JR),I
RETURN
FORMAT I5,A,14, 1,X6,E15.5I
PRINT SR(S)=
SR
SR13)
S
SRITEDMMII)
GO TO 750
SR
SRIO=. F0(1,SA93
OM(IJ):=0DMM(I,J)MOMMM(IK)IDINIJ,K)
DO 500 K=,3
DMM(I). O.SAMO05
0
DO
DO
DO 400
610RD
5(IA(32),
41STR),(IA(2 ),IELT),(IA(29).ITEM),(IA(130) ITIC),IIA131),I FTI, F'SAMOll
21H),(I
IF IT IS
IF
IF IBST-TEST
TESTE7H A
EOUIVALENCE
OIMENSI(N NEL(20,17),MAC(4.4.20),IWGI9DI,SOE'.)I,A9O,7) ,0(,B), ESAM04O3
FOUIVALENCE (IA1349),NTIC). IA1348),ISDT)1(IA(347), ISDY),(11346) F4SAM026
EQUIVALENCE
EQUIVALENCEIlAAA),I(021,33),(02110).F2),ID
COMMON
SUBROUTINE
Table VII-75. Source program listing of subroutine SAME (Link 4)
* LABEL
C SUBROUTINE SAME
DMM(I,21)RED(4)51=MMi2,2) RFSAM094
DMM(SCAL=CIR )*DIR )+CIRI2) IR 2 CIRI )DIR(3)
Table VII-76. Source program listing of function SCAL (Link 4)
* LABEL
C FUNCTION SCAL(CIR,RED)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
C IF IT IS A SMALL POINT
C IF IT IS A PLATE WITH THE SHIFT IN RED VECTOR
C OBTAIN THE SYMPHETIC PART
C PERFORM THE TRANSFORMATION USING THE DIM MAX
C
SUBROUTINE SAME (Link 4)
FUNCTION SCAL(CIR,RED)
C PERFORMS SCALAR VECTOR PRODUCT
C TO OBTAIN THE SCALAR PRODUCT OF VECTORS CIR AND RED IN SCAL
C DIMENSION CIR(3),RED(3)
SCAL(&CIR(1),RED(1))=SCAL(&CIR(2),RED(2))=SCAL(&CIR(3),RED(3))
END
C
Table VII-78. Source program listing of subroutine STRA (Link 4)

* LABEL
C    COORDINATE TRANSFORMATION FOR STS
COMMON 14,4
D     END COORD

Table VII-79. Source program listing of subroutine STRS (Link 4)

* LABEL
C    SUBROUTINE STS
C
C    COMMON 14,AA
D     END COORD
### Table VII-80. Source program listing of subroutine TEMP (Link 4)

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><em>LEVEL</em></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>CENTER</strong></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>C COMPUTES</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>CE4TMP</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>C DIMENSION</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>C LENGTH CHANGE PER UNIT TEMPERATURE PER UNIT DISTANCE</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>C JPL TECHNICAL REPORT 32-1240</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>C TICK NT ON TICK ON TICK TICK ON TICK</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9 RSIR)SR1.I)(K F 14TP405 ETM=IAIJM-)1O00 *PRS F4T055</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>41STR)!</td>
</tr>
</tbody>
</table>
| 11   |      | 21M),(VA(03 ,I ,(
| 12   |      | (174).I(C.II AI75),IDE ) |
| 13   |      | 911A), (IA(63),IDT) |
| 14   |      | 7 (ICZ). |
| 15   |      | 2MSE)       |
| 16   |      | 3(AA(192),N)(AA(2148),AA(147)1G |
| 17   |      | EUIVALENCE |
| 18   |      | CAL 5      |
| 19   |      | ENTRY TICK |
| 20   |      | TICK 4      |
| 21   |      | LINK 4      |
| 22   |      | PZF         |
| 23   |      | IA(137),MZ )1(A1336 ),JPIFI,(IA(335) ITAS1).(IA334),1I) 4TP030 4,) ()333),IP |
| 24   |      | D),(IA( ),3ACEL |
| 25   |      | 401),JARE,
| 26   |      | (AA(171) |
| 27   |      | O),IONE( (20 |
| 28   |      | 15(481.X.IAA(1631,Y) (AA(171) |
| 29   |      | IIJ (AA23).SIRI, 1(AA226),DIN),(AA(23 ,S |
| 30   |      | E 4TP020 3. |
| 31   |      | 771IIS) E4TP020 3. |
| 32   |      | listing of |
| 33   |      | subroutine TEMP (Link 4) |

### Table VII-82. Source program listing of subroutine TOPO (Link 4)

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><em>LEVEL</em></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>CENTER</strong></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>C COMPUTES</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>CE4TMP</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>C DIMENSION</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>C LENGTH CHANGE PER UNIT TEMPERATURE PER UNIT DISTANCE</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>C JPL TECHNICAL REPORT 32-1240</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>C TICK NT ON TICK ON TICK ON TICK</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9 RSIR)SR1.I)(K F 14TP405 ETM=IAIJM-)1O00 *PRS F4T055</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>41STR)!</td>
</tr>
</tbody>
</table>
| 11   |      | 21M),(VA(03 ,I ,(
| 12   |      | (174).I(C.II AI75),IDE ) |
| 13   |      | 911A), (IA(63),IDT) |
| 14   |      | 7 (ICZ). |
| 15   |      | 2MSE)       |
| 16   |      | 3(AA(192),N)(AA(2148),AA(147)1G |
| 17   |      | EUIVALENCE |
| 18   |      | CAL 5      |
| 19   |      | ENTRY TICK |
| 20   |      | TICK 4      |
| 21   |      | LINK 4      |
| 22   |      | PZF         |
| 23   |      | IA(137),MZ )1(A1336 ),JPIFI,(IA(335) ITAS1).(IA334),1I) 4TP030 4,) ()333),IP |
| 24   |      | D),(IA( ),3ACEL |
| 25   |      | 401),JARE,
| 26   |      | (AA(171) |
| 27   |      | O),IONE( (20 |
| 28   |      | 15(481.X.IAA(1631,Y) (AA(171) |
| 29   |      | IIJ (AA23).SIRI, 1(AA226),DIN),(AA(23 ,S |
| 30   |      | E 4TP020 3. |
| 31   |      | 771IIS) E4TP020 3. |
| 32   |      | listing of |
| 33   |      | subroutine TOPO (Link 4) |

### Table VII-81. Source program listing of subroutine TICK (Link 4)

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td><em>LEVEL</em></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td><strong>CENTER</strong></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>C COMPUTES</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>CE4TMP</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>C DIMENSION</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>C LENGTH CHANGE PER UNIT TEMPERATURE PER UNIT DISTANCE</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>C JPL TECHNICAL REPORT 32-1240</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>C TICK NT ON TICK ON TICK ON TICK</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>9 RSIR)SR1.I)(K F 14TP405 ETM=IAIJM-)1O00 *PRS F4T055</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>41STR)!</td>
</tr>
</tbody>
</table>
| 11   |      | 21M),(VA(03 ,I ,(
| 12   |      | (174).I(C.II AI75),IDE ) |
| 13   |      | 911A), (IA(63),IDT) |
| 14   |      | 7 (ICZ). |
| 15   |      | 2MSE)       |
| 16   |      | 3(AA(192),N)(AA(2148),AA(147)1G |
| 17   |      | EUIVALENCE |
| 18   |      | CAL 5      |
| 19   |      | ENTRY TICK |
| 20   |      | TICK 4      |
| 21   |      | LINK 4      |
| 22   |      | PZF         |
| 23   |      | IA(137),MZ )1(A1336 ),JPIFI,(IA(335) ITAS1).(IA334),1I) 4TP030 4,) ()333),IP |
| 24   |      | D),(IA( ),3ACEL |
| 25   |      | 401),JARE,
| 26   |      | (AA(171) |
| 27   |      | O),IONE( (20 |
| 28   |      | 15(481.X.IAA(1631,Y) (AA(171) |
| 29   |      | IIJ (AA23).SIRI, 1(AA226),DIN),(AA(23 ,S |
| 30   |      | E 4TP020 3. |
| 31   |      | 771IIS) E4TP020 3. |
| 32   |      | listing of |
| 33   |      | subroutine TICK (Link 4) |
Table VII-83. Source program listing of subroutine TRAN (Link 4)

```plaintext
C TO C PERFORMS VECTORIAL
C TRANSFORMATION FOR VECTORS
C
C IF positive, change direction and replace with unity, otherwise
C replace with negative of vector and return unity
C
DIMENSION CIR(3)
END
```

Table VII-84. Source program listing of subroutine UNIT (Link 4)

```plaintext
C LABEL
UNIT
SUBROUTINE UNIT(CIR)
DIMENSION CIR(3)
END
```

Table VII-85. Source program listing of subroutine VECT (Link 4)

```plaintext
C TO C PERFORMS VECTORIAL
C TRANSFORMATION FOR VECTORS
C
C IF negative, change direction and replace with unity, otherwise
C replace with negative of vector and return unity
C
DIMENSION CIR(3)
END
```
References


