MEMORANDUM FOR Joint Technical Coordinating Group for Munitions Effectiveness Product Management Office (OC-ALC/ENLB/Ms. Sandra Hysell), 7851 Arnold Street, Suite 202, Tinker AFB, OK 73145-9160

SUBJECT: Distribution Statement for 61 JTCG/ME-71-7-1, 61 JTCG/ME-7-2-1 and 61 JTCG/ME-71-7-2-2

1. A review of the subject Magic Computer Simulation User and Analyst Manuals has been conducted based upon a request received from the US Army Research Laboratory. This review resulted in the decision to release these publications into the public domain. As such, request the following distribution statement be added to these items: “Approved for public release; distribution is unlimited.”

2. Request, therefore, recipients of these publications be notified of this distribution statement.

3. The JTCG/ME Program Office point of contact for this request is Mrs. Chantal B. Marus, COMM (410) 278-6740, DSN 298-6740; e-mail: chantal.b.marus@us.army.mil.

BRYAN W. PARIS
Director
JTCG/ME Program Office
The MAGIC computer simulation generates target description data consisting of item-by-item listings of the target’s components and air-spaces encountered by a large number of parallel rays emanating from any desired attack angle. A combinatorial geometry technique, which defines the locations and shapes of the various physical regions in terms of the intersections and unions of the volumes contained in a set of simple bodies, is used to represent complex target structures. A grid cell pattern is superimposed over the surface of the target and parallel rays are "fired" from each grid cell.
JTCG/ME
MAGIC COMPUTER SIMULATION
VOLUME II. ANALYST MANUAL
PART II

Produced for:
Joint Technical Coordinating Group for Munitions Effectiveness

MAY 1971

Approved for public release; distribution is unlimited.
ABSTRACT

The MAGIC computer simulation generates target description data consisting of item-by-item listings of the target's components and air-spaces encountered by a large number of parallel rays emanating from any desired attack angle. A combinatorial geometry technique, which defines the locations and shapes of the various physical regions in terms of the intersections and unions of the volumes contained in a set of simple bodies, is used to represent complex target structures. A grid cell pattern is superimposed over the surface of the target and parallel rays are "fired" from each grid cell.

Volume II, Part II contains:

Section III  Simulation Model, Subroutine QRTIC through Subroutine TESTG; and List of Symbols and Abbreviations.

Section IV  Source Listing
# TABLE OF CONTENTS

## Part I

### I

**INTRODUCTION** .................................................. 1

**COMBINATORIAL GEOMETRY TECHNIQUE** .................................. 1

**GEOMETRICAL DESCRIPTION** ........................................ 5

- Coordinate System .................................................. 5
- Rectangular Parallelepipeds (RPP) .................................. 5
- Identification Codes ................................................. 5
- Grid .................................................................. 5
- Cellular Output ..................................................... 9
- Data Input Error ................................................... 9

**OPTIONAL ROUTINES** .................................................. 14

- Subroutine TESTG ..................................................... 14
- Subroutine VOLUM ..................................................... 14
- Subroutine AREA ....................................................... 14

### II

**MATHMATICAL MODEL** ................................................ 15

- RECTANGULAR PARALLELEPIPED (SUBROUTINE RPP) ..................... 16
- BOX (SUBROUTINE BOX) ............................................... 19
- SPHERE (SUBROUTINE SPH) ........................................... 22
- RIGHT CIRCULAR CYLINDER (SUBROUTINE RCC) ....................... 25
- RIGHT ELLIPTIC CYLINDER (SUBROUTINE REC) ....................... 29
- RIGHT TRUNCATED CONE (SUBROUTINE TRC) ......................... 32
- ELLIPSOID OF REVOLUTION (SUBROUTINE ELL) ....................... 39
- RIGHT ANGLE WEDGE (SUBROUTINE RAW) .................................. 43
- ARBITRARY POLYHEDRON (SUBROUTINE ARB) ............................ 48
- TRUNCATED ELLIPTIC CONE (SUBROUTINE TEC) ....................... 51
- TORUS (SUBROUTINE TOR) .......................................... 61
- ARBITRARY SURFACE (SUBROUTINE ARS) .................................. 64
- ANGLE OF OBLIQUITY (SUBROUTINE CALC) .............................. 67

Normal to the Rectangular Parallelepipeded ................................... 67
Normal to the Box .................................................................. 68
Normal to the Sphere ...................................................... 68
TABLE OF CONTENTS (Continued)

Section

II (cont'd)

<table>
<thead>
<tr>
<th>Part I (Continued)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal to a Planar Surface of the Right Circular Cylinder, the Right Elliptical</td>
<td>69</td>
</tr>
<tr>
<td>Cylinder, or the Truncated Right Cone</td>
<td></td>
</tr>
<tr>
<td>Normal to the Side of the Right Circular Cylinder</td>
<td>70</td>
</tr>
<tr>
<td>Normal to the Side of the Right Elliptic Cylinder</td>
<td>71</td>
</tr>
<tr>
<td>Normal to the Side of the Truncated Right Cone</td>
<td>73</td>
</tr>
<tr>
<td>Normal to the Ellipsoid of Revolution</td>
<td>74</td>
</tr>
<tr>
<td>Normal to the Slanted Surface of the Right Angle Wedge</td>
<td>75</td>
</tr>
<tr>
<td>Normal to the Arbitrary Polyhedron</td>
<td>76</td>
</tr>
<tr>
<td>Normal to the Truncated Elliptic Cone</td>
<td>77</td>
</tr>
<tr>
<td>Normal to the Torus</td>
<td>81</td>
</tr>
<tr>
<td>Normal to the Arbitrary Surface</td>
<td>82</td>
</tr>
<tr>
<td>Angle of Obliquity</td>
<td>83</td>
</tr>
<tr>
<td>GRID PLANE AND TARGET (SUBROUTINES GRID AND AREA)</td>
<td>85</td>
</tr>
<tr>
<td>SOLUTION OF QUARTIC EQUATION (SUBROUTINE QRTIC)</td>
<td>91</td>
</tr>
<tr>
<td>SOLUTION OF CUBIC EQUATION (SUBROUTINE CUBIC)</td>
<td>93</td>
</tr>
<tr>
<td>THE MASTER-ASTER ARRAY</td>
<td>95</td>
</tr>
<tr>
<td>PROGRAM MAGIC (MAIN PROGRAM)</td>
<td>107</td>
</tr>
<tr>
<td>INPUT PROCESSING (SUBROUTINE GENI)</td>
<td>111</td>
</tr>
<tr>
<td>INPUT OF RECTANGULAR PARALLELEPTIPED (SUBROUTINE RPPIN)</td>
<td>120</td>
</tr>
<tr>
<td>INPUT OF ARBITRARY POLYHEDRON (SUBROUTINE ALBERT)</td>
<td>123</td>
</tr>
<tr>
<td>INPUT OF ARBITRARY SURFACE (SUBROUTINE ARIN)</td>
<td>129</td>
</tr>
<tr>
<td>GRID/ATTACK PLANE (SUBROUTINE GRID)</td>
<td>132</td>
</tr>
<tr>
<td>RAY TRACKING (SUBROUTINE TRACK)</td>
<td>135</td>
</tr>
<tr>
<td>NORMAL AND ANGLE OF OBLIQUITY CALCULATIONS (SUBROUTINE CALC)</td>
<td>140</td>
</tr>
<tr>
<td>RAY INTERSECTIONS (SUBROUTINE G1)</td>
<td>143</td>
</tr>
<tr>
<td>REGION LOCATION (SUBROUTINE WOW!)</td>
<td>146</td>
</tr>
<tr>
<td>LIST OF SYMBOLS AND ABBREVIATIONS</td>
<td>150</td>
</tr>
<tr>
<td>III</td>
<td></td>
</tr>
<tr>
<td>SIMULATION MODEL</td>
<td>163</td>
</tr>
<tr>
<td>PROGRAM MAGIC - MAIN ROUTINE</td>
<td>165</td>
</tr>
<tr>
<td>Subroutine GENI</td>
<td>175</td>
</tr>
<tr>
<td>Subroutine RPPIN</td>
<td>176</td>
</tr>
<tr>
<td>Subroutine ALBERT</td>
<td>208</td>
</tr>
<tr>
<td>Subroutine ARIN</td>
<td>218</td>
</tr>
<tr>
<td>Subroutine ARIN</td>
<td>226</td>
</tr>
<tr>
<td>Subroutine SEE3</td>
<td>231</td>
</tr>
<tr>
<td>Subroutine GRID</td>
<td>234</td>
</tr>
</tbody>
</table>

iv
### TABLE OF CONTENTS (Continued)

#### Part I (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>III (cont'd)</td>
<td>Subroutine TRACK</td>
<td>242</td>
</tr>
<tr>
<td></td>
<td>Subroutine CALC</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>Subroutine G1</td>
<td>289</td>
</tr>
<tr>
<td></td>
<td>Subroutine WOWI</td>
<td>311</td>
</tr>
<tr>
<td></td>
<td>Subroutine RPP</td>
<td>320</td>
</tr>
<tr>
<td></td>
<td>Subroutine BOX</td>
<td>328</td>
</tr>
<tr>
<td></td>
<td>Subroutine SPH</td>
<td>335</td>
</tr>
<tr>
<td></td>
<td>Subroutine RCC</td>
<td>339</td>
</tr>
<tr>
<td></td>
<td>Subroutine REC</td>
<td>352</td>
</tr>
<tr>
<td></td>
<td>Subroutine TRC</td>
<td>367</td>
</tr>
<tr>
<td></td>
<td>Subroutine ELL</td>
<td>384</td>
</tr>
<tr>
<td></td>
<td>Subroutine RAW</td>
<td>389</td>
</tr>
<tr>
<td></td>
<td>Subroutine ARB</td>
<td>400</td>
</tr>
<tr>
<td></td>
<td>Subroutine TEC</td>
<td>407</td>
</tr>
<tr>
<td></td>
<td>Subroutine ARS</td>
<td>430</td>
</tr>
<tr>
<td></td>
<td>Subroutine TOR</td>
<td>443</td>
</tr>
</tbody>
</table>

#### Part II

<table>
<thead>
<tr>
<th>Section</th>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>III (cont'd)</td>
<td>Subroutine QRTIC</td>
<td>453</td>
</tr>
<tr>
<td></td>
<td>Subroutine CUBIC</td>
<td>459</td>
</tr>
<tr>
<td></td>
<td>Subroutine UN2</td>
<td>464</td>
</tr>
<tr>
<td></td>
<td>Subroutine UN3</td>
<td>465</td>
</tr>
<tr>
<td></td>
<td>Subroutine OPENK</td>
<td>466</td>
</tr>
<tr>
<td></td>
<td>Subroutine RAN</td>
<td>467</td>
</tr>
<tr>
<td></td>
<td>Subroutine URAN 31</td>
<td>468</td>
</tr>
<tr>
<td></td>
<td>Subroutine CROSS</td>
<td>469</td>
</tr>
<tr>
<td></td>
<td>Subroutine DOT</td>
<td>470</td>
</tr>
<tr>
<td></td>
<td>Subroutine UNIT</td>
<td>471</td>
</tr>
<tr>
<td></td>
<td>Subroutine XDIST</td>
<td>472</td>
</tr>
<tr>
<td></td>
<td>Subroutine DCOSP</td>
<td>473</td>
</tr>
<tr>
<td></td>
<td>Subroutine TROPIC</td>
<td>474</td>
</tr>
<tr>
<td></td>
<td>Function S</td>
<td>476</td>
</tr>
<tr>
<td></td>
<td>Subroutine RPP2</td>
<td>477</td>
</tr>
<tr>
<td></td>
<td>Subroutine VOLUM</td>
<td>481</td>
</tr>
<tr>
<td></td>
<td>Subroutine AREA</td>
<td>495</td>
</tr>
<tr>
<td></td>
<td>Subroutine TESTG</td>
<td>505</td>
</tr>
</tbody>
</table>

**LIST OF SYMBOLS AND ABBREVIATIONS** | 509

#### IV

**SOURCE LISTING** | 609
**SIMULATION SOURCE DECK** | 609
**SAMPLE PROBLEM DECK** | 609
**LIST OF FIGURES**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Part I</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Mallet with Handle and Head as Separate Regions</td>
</tr>
<tr>
<td>2</td>
<td>Mallet with Handle Extending Into the Head</td>
</tr>
<tr>
<td>3</td>
<td>Mallet with Handle Consisting of Two Types of Materials</td>
</tr>
<tr>
<td>4</td>
<td>Mallet with Head and Handle of Like Materials</td>
</tr>
<tr>
<td>5</td>
<td>Buttressing Surfaces</td>
</tr>
<tr>
<td>6</td>
<td>Coordinate System Superimposed on Simplified Tank</td>
</tr>
<tr>
<td>7</td>
<td>Twelve RPF's</td>
</tr>
<tr>
<td>8</td>
<td>Attack Angle Geometry</td>
</tr>
<tr>
<td>9</td>
<td>Representative Vehicle Section for Target Cell Description Data</td>
</tr>
<tr>
<td>10</td>
<td>Rectangular Parallelepiped</td>
</tr>
<tr>
<td>11</td>
<td>Box</td>
</tr>
<tr>
<td>12</td>
<td>Sphere</td>
</tr>
<tr>
<td>13</td>
<td>Right Circular Cylinder</td>
</tr>
<tr>
<td>14</td>
<td>Right Elliptic Cylinder</td>
</tr>
<tr>
<td>15</td>
<td>Right Truncated Cone</td>
</tr>
<tr>
<td>16</td>
<td>Ellipsoid of Revolution</td>
</tr>
<tr>
<td>17</td>
<td>Right Angle Wedge</td>
</tr>
<tr>
<td>18</td>
<td>Right Angle Wedge Normal Derivation</td>
</tr>
<tr>
<td>19</td>
<td>Arbitrary Polyhedron</td>
</tr>
<tr>
<td>20</td>
<td>Truncated Elliptic Cone</td>
</tr>
<tr>
<td>21</td>
<td>TEC Cross-Section</td>
</tr>
<tr>
<td>22</td>
<td>Torus</td>
</tr>
<tr>
<td>23</td>
<td>Arbitrary Surface</td>
</tr>
<tr>
<td>24</td>
<td>ARS Surface Triangle</td>
</tr>
<tr>
<td>25</td>
<td>Rectangular Parallelepiped Box</td>
</tr>
<tr>
<td>26</td>
<td>Sphere</td>
</tr>
<tr>
<td>27</td>
<td>Right Circular Cylinder</td>
</tr>
<tr>
<td>28</td>
<td>Right Elliptic Cylinder</td>
</tr>
<tr>
<td>29</td>
<td>Truncated Right Cone</td>
</tr>
<tr>
<td>30</td>
<td>Ellipsoid of Revolution</td>
</tr>
<tr>
<td>31</td>
<td>Right Angle Wedge</td>
</tr>
<tr>
<td>32</td>
<td>Arbitrary Polyhedron Truncated Elliptic Cone TEC Cross-Section</td>
</tr>
<tr>
<td>33</td>
<td>TEC Intersection Ellipse</td>
</tr>
<tr>
<td>34</td>
<td>Torus</td>
</tr>
<tr>
<td>35</td>
<td>ARS Surface Triangle</td>
</tr>
<tr>
<td>36</td>
<td>Grid Plane Geometry</td>
</tr>
<tr>
<td>37</td>
<td>Example of a Grid Plane</td>
</tr>
<tr>
<td>38</td>
<td>Example Grid Cell</td>
</tr>
<tr>
<td>39</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>42</td>
<td>97</td>
</tr>
<tr>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>44</td>
<td>102</td>
</tr>
<tr>
<td>45</td>
<td>103</td>
</tr>
<tr>
<td>46</td>
<td>104</td>
</tr>
<tr>
<td>47</td>
<td>105</td>
</tr>
<tr>
<td>48</td>
<td>106</td>
</tr>
<tr>
<td>49</td>
<td>109</td>
</tr>
<tr>
<td>50</td>
<td>116</td>
</tr>
<tr>
<td>51</td>
<td>122</td>
</tr>
<tr>
<td>52</td>
<td>127</td>
</tr>
<tr>
<td>53</td>
<td>131</td>
</tr>
<tr>
<td>54</td>
<td>134</td>
</tr>
<tr>
<td>55</td>
<td>139</td>
</tr>
<tr>
<td>56</td>
<td>142</td>
</tr>
<tr>
<td>57</td>
<td>145</td>
</tr>
<tr>
<td>58</td>
<td>149</td>
</tr>
<tr>
<td>59</td>
<td>164</td>
</tr>
<tr>
<td>60</td>
<td>320</td>
</tr>
<tr>
<td>61</td>
<td>328</td>
</tr>
<tr>
<td>62</td>
<td>335</td>
</tr>
<tr>
<td>63</td>
<td>339</td>
</tr>
<tr>
<td>64</td>
<td>352</td>
</tr>
<tr>
<td>65</td>
<td>367</td>
</tr>
<tr>
<td>66</td>
<td>384</td>
</tr>
<tr>
<td>67</td>
<td>389</td>
</tr>
<tr>
<td>68</td>
<td>400</td>
</tr>
<tr>
<td>69</td>
<td>407</td>
</tr>
<tr>
<td>70</td>
<td>430</td>
</tr>
<tr>
<td>71</td>
<td>443</td>
</tr>
<tr>
<td>72</td>
<td>444</td>
</tr>
<tr>
<td>73</td>
<td>481</td>
</tr>
<tr>
<td>74</td>
<td>610</td>
</tr>
<tr>
<td>75</td>
<td>614</td>
</tr>
<tr>
<td>76</td>
<td>625</td>
</tr>
<tr>
<td>77</td>
<td>628</td>
</tr>
<tr>
<td>78</td>
<td>632</td>
</tr>
</tbody>
</table>

**Part II**

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>73</td>
<td>481</td>
</tr>
<tr>
<td>74</td>
<td>610</td>
</tr>
<tr>
<td>75</td>
<td>614</td>
</tr>
<tr>
<td>76</td>
<td>625</td>
</tr>
<tr>
<td>77</td>
<td>628</td>
</tr>
<tr>
<td>78</td>
<td>632</td>
</tr>
</tbody>
</table>

viii
<table>
<thead>
<tr>
<th>Figure</th>
<th>Source Listing, Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>79</td>
<td>SEE3</td>
<td>633</td>
</tr>
<tr>
<td>80</td>
<td>GRID</td>
<td>634</td>
</tr>
<tr>
<td>81</td>
<td>TRACK</td>
<td>637</td>
</tr>
<tr>
<td>82</td>
<td>CALC</td>
<td>641</td>
</tr>
<tr>
<td>83</td>
<td>G1</td>
<td>651</td>
</tr>
<tr>
<td>84</td>
<td>WOWI</td>
<td>658</td>
</tr>
<tr>
<td>85</td>
<td>RPP</td>
<td>661</td>
</tr>
<tr>
<td>86</td>
<td>BOX</td>
<td>663</td>
</tr>
<tr>
<td>87</td>
<td>SPH</td>
<td>665</td>
</tr>
<tr>
<td>88</td>
<td>RCC</td>
<td>666</td>
</tr>
<tr>
<td>89</td>
<td>REC</td>
<td>669</td>
</tr>
<tr>
<td>90</td>
<td>TRC</td>
<td>672</td>
</tr>
<tr>
<td>91</td>
<td>ELL</td>
<td>676</td>
</tr>
<tr>
<td>92</td>
<td>RAW</td>
<td>677</td>
</tr>
<tr>
<td>93</td>
<td>ARB</td>
<td>680</td>
</tr>
<tr>
<td>94</td>
<td>TEC</td>
<td>682</td>
</tr>
<tr>
<td>95</td>
<td>ARS</td>
<td>687</td>
</tr>
<tr>
<td>96</td>
<td>TOR</td>
<td>693</td>
</tr>
<tr>
<td>97</td>
<td>QRTIC</td>
<td>696</td>
</tr>
<tr>
<td>98</td>
<td>CUBIC</td>
<td>698</td>
</tr>
<tr>
<td>99</td>
<td>UN2</td>
<td>699</td>
</tr>
<tr>
<td>100</td>
<td>UN3</td>
<td>700</td>
</tr>
<tr>
<td>101</td>
<td>OPENK</td>
<td>701</td>
</tr>
<tr>
<td>102</td>
<td>RAN</td>
<td>702</td>
</tr>
<tr>
<td>103</td>
<td>URAN31</td>
<td>703</td>
</tr>
<tr>
<td>104</td>
<td>CROSS</td>
<td>704</td>
</tr>
<tr>
<td>105</td>
<td>DOT</td>
<td>705</td>
</tr>
<tr>
<td>106</td>
<td>UNIT</td>
<td>706</td>
</tr>
<tr>
<td>107</td>
<td>XDIST</td>
<td>707</td>
</tr>
<tr>
<td>108</td>
<td>DCOSP</td>
<td>708</td>
</tr>
<tr>
<td>109</td>
<td>TROPIC</td>
<td>709</td>
</tr>
<tr>
<td>110</td>
<td>S</td>
<td>710</td>
</tr>
<tr>
<td>111</td>
<td>RPP2</td>
<td>711</td>
</tr>
<tr>
<td>112</td>
<td>VOLUM</td>
<td>712</td>
</tr>
<tr>
<td>113</td>
<td>AREA</td>
<td>716</td>
</tr>
<tr>
<td>114</td>
<td>TESTG</td>
<td>719</td>
</tr>
<tr>
<td>115</td>
<td>Sample Problem Data Deck</td>
<td>720</td>
</tr>
</tbody>
</table>
Subroutine QRTIC(C,R,N)

Subroutine QRTIC is called by subroutine TOR to solve a polynomial equation of the type \( x^4 + ax^3 + bx^2 + cx + d = 0 \), using Ferrari's solution of the quartic equation where the coefficient of \( x^4 \) is assumed to be one. Four-element array R will contain the roots, and variable N will contain the number of real roots. If there are two real roots, they will be placed in R(1) and R(2) with the complex roots in R(3) and R(4) in the form \( R(3) \pm iR(4) \). If there are no real roots, the complex roots will be stored in the form \( R(1) \pm R(2) \) and \( R(2) \pm iR(4) \). Subroutine QRTIC calls subroutine CUBIC to find the roots of the resolvent cubic equation.

The statement

```plaintext
DIMENSION C(4),R(4),CC(3),RR(3)
```

is used to dimension arrays for the four coefficients of the quartic equation, for the four roots of the quartic equation, for the three coefficients of the resolvent cubic equation, and for three roots of the cubic equation.

The statements

```plaintext
SOLVES A POLYNOMIAL EQUATION OF THE TYPE X**4 + C(1)*X**3 + C(2)*X**2 + C(3)*X + C(4) = 0 USING THE FERRARI SOLUTION OF THE QUARTIC EQUATION. THE COEFFICIENT OF X**4 IS ASSUMED TO BE 1. R(4) CONTAINS THE ROOTS, N CONTAINS THE NUMBER OF REAL ROOTS. IF THERE ARE 2 REAL ROOTS THEY WILL BE IN R(1) AND R(2), WITH THE COMPLEX ROOTS IN R(3) = R(4)*I. IF THERE ARE NO REAL ROOTS THE COMPLEX ROOTS ARE IN R(1) = R(2)*I AND R(3) = R(4)*I.

COMPUTE RESOLVENT CUBIC

C150=C(1)*C(1)
CC(1)=.5*C(2)
CC(2)=.25*C(1)*C(3)-C(4)
CC(3)=.125*(C(4)^2+C(2)-C150-C(3))*C(3)
CALL CUBIC(CC,RR,NN)
```

are used to compute the coefficients of the resolvent cubic equation

\[
\begin{align*}
4w^3 - \frac{b}{2}w^2 + \left(\frac{ac-4d}{4}\right)w + \frac{4bd-a^2d-c^2}{8} = 0
\end{align*}
\]

where \( y = 2w \), from the coefficients of the quartic equation. Subroutine CUBIC is called to compute the roots of the resolvent cubic equation.
The statement

```c
C DETERMINE IF POSSIBLE SOLUTION
C T=.25*CE0-C(2)
```

is used to compute the quantity \( a^2/4-b \) for use in solving the coefficients of

\[
\left(\frac{a^2}{2}-b+2w\right)x^2 + (aw-c) x + (w^2-d) = e^2 x^2 + 2efx + f^2
\]

which splits into two quadratic equations with properly chosen \( e \) and \( f \):

\[
x^2 + \frac{a}{2} x + \frac{1}{2} y = \pm (ex+f)
\]

Solving for \( x \) using the quadratic formula results in

\[
x_{1,2} = \frac{-\frac{a}{2} + \frac{e}{2} \pm \sqrt{\left(-\frac{a}{2} + \frac{e}{2}\right)^2 - 4\left(w-f\right)}}{2}
\]

for two roots and

\[
x_{3,4} = \frac{-\frac{a}{2} - \frac{e}{2} \pm \sqrt{\left(-\frac{a}{2} - \frac{e}{2}\right)^2 - 4\left(w-f\right)}}{2}
\]

for the remaining two roots.

The statement

```c
DO 10 I=1*NN
```

is used to begin a DO loop to test each real root returned from Subroutine CUBIC to determine if there are any real roots in the quartic equation.

The statements

```c
ROOT#RR(I)
ASQ=T+ROOT*ROOT
IF(ABS(ASQ)+LE.0.000001)ASQ=0.
IF(ASQ+LT.0)GOTO 10
```
are used to compute the coefficient of $x^2$ of Equation (202). If it is very nearly zero, it is set to zero. The coefficient is tested for a less-than-zero condition, which means that for the given real root of the cubic equation there are no real roots for the quartic equation. Therefore, the program loops to consider the next real root, if any.

The statements

```plaintext
RSQ=ROOT$ROOT=C(4)
IF(ABS(BSQ)$LE.0.000001)BSQ=0.
IF(BSQ,GE.0.0)GOTO 20
10 CONTINUE
N=0
RETURN
```

are used to compute the constant term of Equation (202). If it is very nearly zero it is set to zero. The value is tested for a greater than or equal to zero condition, which means that for the given real root of the cubic equation there are real roots for the quartic equation. If the constant term of Equation (202) is less than zero, the given real root of the cubic equation will not result in real roots for the quartic equation. The program therefore loops to consider the next real cubic root. If all real cubic roots have been considered without satisfying the conditions for real roots in the quartic equation, the variable N (the number of real roots in the quartic equation) is set to zero, and the subroutine returns control to Subroutine TOR.

The statements

```plaintext
C  COMPUTE FIRST TWO ROOTS OF QUARTIC EQUATION
 20 TWOAB=C(1)+ROOT=C(3)
  A=SORT(ASQ)
  R=STON(SORT(BSQ)$TWOAB)
  N=N
```

are used to compute the coefficient of $x$, the square root of the coefficient of $x^2$, and the square root of the constant term with the sign of the coefficient of $x$ of Equation (202). The variable N (the number of real roots in the quartic equation) is initialized to zero.
The statements

```
REAL*8 A,B,C(3)
DISC=REAL*8 REAL=ROOT+B
SQRT=SQRT(ABS(DISC))
```

are used to compute the values for solving quadratic Equation (203) for the first set of roots; and to solve for the value of the square root of the discriminate.

The statements

```
IF (ABS(DISC) LE 0.000001) DISC=0.
IF (DISC LT 0.0) GOTO 30
```

are used to determine if the absolute value of the discriminate is very nearly equal to zero, and, if it is, to set it to zero. A test is made to determine if the value of the discriminate is less than zero. If it is, the roots are complex conjugates. Therefore, the program branches to compute the two imaginary roots.

The statements

```
C4 DISCRIMINATE GE 0 COMPUTE 2 REAL ROOTS
C
N=2
R(1)=REAL=SQRT
R(2)=REAL=SQROOT
GOTO 40
```

are executed if the discriminate was greater than or equal to zero. They are used to set variable N to two to indicate two real roots. The values of the roots are computed and stored in the first two elements of array R. The program then branches to determine the condition of the two remaining roots.

The statements

```
C5 DISCRIMINATE LT 0 COMPUTE 2 IMAGINARY ROOTS
C
30 R(3)=REAL
R(4)=SQROOT
```
are executed if the discriminate was less than zero, indicating that the roots are complex conjugates. These statements therefore store the imaginary roots in the last two elements of array \( R \).

The statements

```c
COMPUTE LAST TWO ROOTS OF QUARTIC EQUATION

REAL=REAL=A
DISC=REAL*REAL-ROOT-B
SOROOT=SQR(ABS(DISC))
```

are used to compute the values for solving quadratic Equation (204) for the second set of roots and to solve for the value of the square root of the discriminate.

The statements

```c
IF(ABS(DISCS)*(LE.*0.000001)*DISC=0.*
IF(DISCS*(LT.*0.0)GOTO 50
```

are used to determine if the absolute value of the discriminate is very nearly equal to zero, and, if it is, to set it to zero. A test is made to determine if the value of the discriminate is less than zero. If it is, the roots are complex conjugates, and the program branches to store the two imaginary roots.

The statements

```c
DISCRIMATE .GE. 0  COMPUTE 2 REAL ROOTS

N=N+2
R(N)=REAL-SOROOT
R(N-1)=REAL+SOROOT
RETURN
```

are executed if the previous test of the discriminate proved that the roots are real. These statements store the two real roots in the \( R \) array after determining the proper location in the \( R \) array based on the number of real roots already computed. Control is then returned to Subroutine TOR with the values of the real roots.
The statements

```c
C
CA  DISCRIMINATE .LT. 0  COMPUTE 2 IMAGINARY ROOTS
C
50  R(N+1)=REAL
    R(N+2)=SQROOT
    RETURN
```

are executed if the discriminate was less than zero, indicating that the roots are complex conjugates. These statements therefore store the imaginary roots in the R array at a location based on the number of real roots already computed. Control is then returned to Subroutine TOR with the values of the real roots.
Subroutine CUBIC(C,R,N)

Subroutine CUBIC is called by Subroutine QRTIC to solve a polynomial equation of the type \( x^3 + ax^2 + bx + c = 0 \) where the coefficient of \( x^3 \) is assumed to be one. The three roots are passed back to Subroutine QRTIC through argument R. The number of real roots is passed back through argument N. If there is only one real root, it will be in R(1) with the complex roots in R(2) and R(3).

The statement

\[
\text{DIMENSION C(3)*R(3)}
\]

is used to dimension two three-element arrays for the three coefficients of the polynomial equation and the three roots of the equation.

The statements

\[
\begin{align*}
  & C1SG=C(1)*C(1), \\
  & P=C(2)-C1SG/3, \\
  & Q=C(3)*C(1)*(2+C1SG/27)=C(2)/3
\end{align*}
\]

are used to solve the expressions (from Cardan's Solution)

\[
\begin{align*}
  p &= b - \frac{a^2}{3} \tag{207} \\
  q &= c + a \left(2\frac{a^2}{27} - \frac{b}{3}\right) \tag{208}
\end{align*}
\]

where \( c(1) = a \), \( c(2) = b \), and \( c(3) = c \)

The statements

\[
\begin{align*}
  & \text{DISC} = -s*pe*p+27*a*q*Q \\
  & C3=C(1)/3
\end{align*}
\]

are used to solve the expression

\[
Q = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2 \tag{214}
\]
from the equation
\[ q = \left( \frac{p}{3} \right)^3 + \left( \frac{q}{2} \right)^2 \]  
and \( a/3 \) from the equation
\[ x = y - \frac{a}{3} \]

The statements

\[
\text{IF} \left( \text{ABS}(\text{DISC}) \leq 1 \times 10^{-8} \right) \text{DISC} = 0.
\]
\[
\text{IF} \left( \text{DISC} \leq 0 \right) \text{GOTO 10}
\]

are used to set \( \text{DISC} \) to zero if the absolute value of \( \text{DISC} \) is very nearly zero. \( \text{DISC} \) is tested to determine if it is less than or equal to zero. If it is, the program branches to compute three real roots for the polynomial equation. If \( \text{DISC} \) is greater than zero, there are only one real root and two complex roots.

The statement

\[
\text{C} \quad \text{CONDITION FOR 1 REAL AND 2 COMPLEX ROOTS}
\]
\[
N = 1
\]

is used to set \( N \) to one to indicate to the calling program that the solution of the cubic equation results in only one real root.

The statements

\[
\text{SQROOT} = \text{SQRT} \left( \frac{\text{DISC}}{108} \right)
\]
\[
\text{HALFQ} = \frac{3}{2}
\]
\[
\text{ACU} = \text{HALFQ} + \text{SQROOT}
\]
\[
\text{BCU} = \text{HALFQ} - \text{SQROOT}
\]

are used to solve the quantities under the radical of equations

\[ A = \frac{3}{2} \sqrt{-rac{q}{2} + \sqrt{q}} \]
and

\[ B = \frac{3}{\sqrt{2} - \sqrt{Q}} \]  

(213)

The statements

\[
A = \text{SIGN}(\text{ABS}(ACU)) \times 3333333333333333 \times ACU
\]

\[
B = \text{SIGN}(\text{ABS}(BCU)) \times 3333333333333333 \times BCU
\]

are used to solve Equations (212) and (213) for A and B, respectively, using the sign of the expression under the radical as the sign of the result.

The statement

\[ AB = A * B \]

is used to compute the value of A + B for solving the equations

\[
y_1 = A + B
\]

\[
y_2,3 = -\frac{A + B}{2} \pm \frac{A - B}{2} \sqrt{3}
\]

The statements

\[
R(1) = AB = C3
\]

\[
R(2) = 5 * AB = C3
\]

\[
R(3) = 866025404 * (A - B)
\]

RETURN

are used to compute the one real root, with the result in R(1), and the two conjugate complex roots, with the real part of the complex root in R(2) and the complex part in R(3) using Equations (212) and (213) and Equation (211). Program control returns to Subroutine QRTIC.

The statement

\[
C3
\]

CONDITION FOR 3 REAL ROOTS

\[
C
\]

10 N=3
is used to set N to three to indicate to the calling program that the solution of the cubic equation results in three real roots.

The statements

\[ T = \sqrt{\frac{p}{3}} \]
\[ T = T + T \]

are used to compute the value of \( 2 \sqrt{-\frac{p}{3}} \) for use in solving the cubic equations (trigonometric solution)

\[ y_1 = 2 \sqrt{-\frac{p}{3}} \cos \left( \frac{\phi}{3} \right) \]  
\[ y_{2,3} = -2 \sqrt{-\frac{p}{3}} \cos \left( \frac{\phi}{3} \pm 60^\circ \right) \]

The statement

\[ \text{IF}(\text{DISC} = 0) \text{GOTO 20} \]

is used to determine if two of the three real roots are equal by testing DISC for a zero value.

The statement

\[ \text{PHI} = \text{ATAN2} \left( \frac{\sqrt{\text{DISC} - 27q^2}}{q} \right) \]

is used to compute the value of \( \frac{\phi}{3} \) in radians for solving Equations (215) and (216) where

\[ \cos \phi = \frac{q}{2 \sqrt{-\frac{(p/3)^3}} \]  

The statements

\[ R(1) = T \times \text{COS} \left( \text{PHI} \right) = C3 \]
\[ R(2) = T \times \text{COS} \left( \text{PHI} + 2.094395103 \right) = C3 \]
\[ R(3) = T \times \text{COS} \left( \text{PHI} + 2.094395103 \right) = C3 \]
\[ \text{RETURN} \]
are used to compute the three unequal roots from Equations (215) and (216) and Equation (217). The program returns control to Subroutine QRTIC.

The statements

```
C
C4 CONDITION FOR 2 OR 3 EQUAL ROOTS
C
20 R(1)=SIGN(TT-Q)=C3
R(2)=SIGN(T-Q)=C3
R(3)=R(2)
RETURN
```

are executed if two of the three real roots are equal. One root is equal to \(2 \sqrt{-\left(\frac{p}{3}\right)} = y\) in Equation (215) with the sign of \(y\) equal to the sign of \(-q\). The other two equal roots are equal to \(\sqrt{-\left(\frac{p}{3}\right)} = y\) in Equation (216) with the sign of \(y\) equal to the sign of \(q\). The program returns control to Subroutine QRTIC.
Subroutine UN2(L,J1,J2)

Subroutine UN2 is called by other subroutines of the MAGIC program when unpacking of two integer data items from a single computer word is required. L, the pointer to the location of the word in the MASTER array, is passed to Subroutine UN2. Subroutine UN2 unpacks the two 15-bit integer data items and passes them back to the calling routine via Subroutine UN2 integer variables J1 and J2. The two data items appear in packed format as:

```
  Unused  |  15 bits  |  15 bits  |
   L     |     J1    |     J2    |
```

The statement

```
C UNPACK 2 15-BIT INTEGER DATA ITEMS FROM L WORD IN MASTER ARRAY
C COMMON MASTER(10000)
```

is used to make the MASTER array available to this subroutine.

The statements

```
I3=MASTER(L)
J1=I3/32768
J2=I3-J1*32768
RETURN
```

are used to store the MASTER array word referenced by pointer L in word I3. Variable J1 is equated to the data in the 15 bits previous to the last 15 bits by dividing the word by $2^{15}$, thus shifting right 15 bits and leaving only J1. The J2 data item is unpacked by subtracting J1 left-shifted by 15 bits from I3, leaving only J2. The left-shift of 15 bits is performed by multiplying the word by $2^{15}$. 
Subroutine UN3(L,J1,J2,J3)

Subroutine UN3 is called by Subroutine Gl when unpacking of three integer data items from L10, Subroutine Gl working storage in the MASTER array is required. L, the location of the packed word, is passed to Subroutine UN3 from Subroutine Gl. Subroutine UN3 unpacks the two six-bit and one 15-bit integer data items and passes them back to Subroutine Gl via Subroutine UN3 integer variables J1, J2, and J3. The three data items appear in packed format as:

<table>
<thead>
<tr>
<th></th>
<th>6 bits</th>
<th>6 bits</th>
<th>15 bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unused</td>
<td>J1</td>
<td>J2</td>
<td>J3</td>
</tr>
</tbody>
</table>

L

The statement

```
UNPACK 2 6-BIT AND 1 15-BIT INTEGER DATA ITEMS FROM G1 WORKING STORAGE AT THE L WORD IN THE MASTER ARRAY
COMMON MASTER(10000)
```

is used to make the MASTER array available to this subroutine.

The statements

```
I3=MASTER(L)
I2=I3/32768
J1=I2/64
J2=I2-J1/64
J3=I3-I2*32768
RETURN
```

are used to store the MASTER array word referenced by pointer L in word I3. I3 is right-shifted 15 bits and equated to I2, leaving only J1 and J2. J1 is determined by right-shifting I2 six bits, leaving only the six bits of J1. J2 is determined by subtracting J1, left-shifted by six bits, from I2, leaving only J2. J3 is finally determined by subtracting J1 and J2 of I2, left-shifted by 15 bits, from I3, leaving only J3. The right-shift of 6 bits or 15 bits is performed by dividing word L by $2^6$ or $2^{15}$, respectively. The left-shift of 6 bits or 15 bits is performed by multiplying word L by $2^6$ or $2^{15}$ respectively.
Subroutine OPENK(L,J1,J2,J3)

Subroutine OPENK is called by Subroutine CALC when unpacking of three integer data items from the ITR array prepared by Subroutine TRACK is required. L, pointer to the location of the word in the ITR array, is passed to this subroutine from Subroutine CALC. Subroutine OPENK unpacks the three 12-bit integer data items and passes them back to Subroutine CALC via integer variables J1, J2, and J3. The three data items appear in packed format as:

```
<table>
<thead>
<tr>
<th>Unused</th>
<th>J1</th>
<th>J2</th>
<th>J3</th>
</tr>
</thead>
</table>
```

The statement
```
COMMON/STRA/ITR,DM,KHIT,LMAX,TR(200),XS(3),IRSTR,ITEM
```

is used to make the ITR array available to this subroutine.

The statements
```
C: UNPACK 3 12-BIT INTEGER DATA ITEMS FROM COMPONENT LINE-OF-SIGHT
C: STORAGE ARRAY ITR. THE THREE ITEMS ARE
C: SURFACE NUMBER / BODY NUMBER / NEXT REGION /

I3=ITR(L)
I2=I3/4096
J1=I2/4096
J2=I2-J1*4096
J3=I3-I2*4096
RETURN
```

are used to store the ITR array word referenced by L in word I3. I3 is right-shifted 12 bits and equated to I2, leaving only J1 and J2 in I2. J1 is determined by right-shifting I2 12 bits, leaving only the 12 bits of J1. J2 is determined by subtracting J1, left-shifted by 12 bits, from I2, leaving only J2. J3 is finally determined by subtracting J1 and J2 of I2, left-shifted by 12 bits, from I3, leaving only J3. The left-shift of word L by 12 bits is performed by multiplying by $2^{12}$. The right-shift is performed by dividing word L by $2^{12}$. 

466
Function RAN(M)

Function RAN is called by Subroutines GRID, AREA, and TROPIC to generate a random number between zero and one.

The statement

```
COMMON/RANDM/IRN
```

is used to pass a number into this routine from which a number between zero and 0.9 will be generated. A new number is passed back into the common area for use when the random function is again called.

The statements

```
GIVEN A NUMBER IRN, GENERATE A RANDOM NUMBER BETWEEN 0 AND 1
RANURAN31(IRN)
RETURN
```

are used to call Function URAN31 to generate a random number from the number given in IRN and to return control to the calling subroutine when the random number has been determined.
Function URAN31(I)

Function URAN31 is called by Function RAN to perform the actual computations of the generating of a random number between zero and 0.9. The argument is also revised.

The statements

```plaintext
IF(I) = 10 * 10 * 10
10 I = 11111111
```

are used to determine if the argument for Functions RAN and URAN31 has a value other than zero. If the argument is zero, the number 11111111 is assigned to the argument.

The statements

```plaintext
20 J = 1
   J = J * 25
   J = J \times (J/67108864) \times 67108864
   J = J * 25
   J = J \times (J/67108864) \times 67108864
   J = J * 25
   J = J \times (J/67108864) \times 67108864
   A1 = J
   I = J
   URAN31 = A1 / 67108864
   RETURN
```

are used to generate a random number from the number 11111111 if the function argument is zero; or from the value of the argument if it is other than zero. The resultant random number between zero and 0.9 is assigned to the function name; at the same time a new value for the argument is assigned for future calls to the random function.
Subroutine CROSS(ANSWER,FIRST,SECOND)

Subroutine CROSS can be called by Subroutine GENI, CALC, or ARS for the purpose of computing the coordinates of the resultant vector from the cross product of two vectors.

The statement

\[
\text{DIMENSION ANSWER(3),FIRST(3),SECOND(3)}
\]

is used to dimension three three-element arrays for the coordinates of the two known vectors and the resultant cross-product vector.

The statements

\[
\begin{align*}
\text{C1} & \quad \text{compute cross product} \\
\text{C2} & \quad \text{ANSWER = FIRST X SECOND} \\
\text{C3} & \quad \text{RETURN}
\end{align*}
\]

are used to compute the cross-product of two vectors whose coordinates are passed by arguments FIRST and SECOND; and to store the coordinates of the resultant vector in argument ANSWER for passing back to the calling program. The cross product of two vectors \( \vec{A} \) and \( \vec{B} \) is given by the expression

\[
\vec{A} \times \vec{B} = (\vec{A} \cdot \vec{B}_z - \vec{A}_z \cdot \vec{B}_y)\hat{x} + (\vec{A}_z \cdot \vec{B}_x - \vec{A}_x \cdot \vec{B}_z)\hat{y} + (\vec{A}_x \cdot \vec{B}_y - \vec{A}_y \cdot \vec{B}_x)\hat{z}
\]
Function DOT(FIRST,SECOND)

Function DOT is called by Subroutines GENI, CALC, ARS, TEC, and TOR for computing the resultant scalar quantity from the dot product of vectors FIRST and SECOND.

The statement

```
DIMENSION FIRST(3), SECOND(3)
```

is used to dimension two three-element arrays for the coordinates of the two vectors passed in the arguments.

The statements

```
COMPUTE DOT PRODUCT

DOT = FIRST : SECOND

RETURN
```

are used to compute the dot product of two vectors whose coordinates are passed by function arguments FIRST and SECOND. The resultant scalar quantity is returned to the calling program via the function name. The dot product of two vectors $\vec{A}$ and $\vec{B}$ is given by the expression

$$\vec{A} \cdot \vec{B} = A_x B_x + A_y B_y + A_z B_z$$
Subroutine UNIT(V)

Subroutine UNIT is called by Subroutine GENI or Subroutine CALC to compute the direction cosines of a vector passed to this subroutine by argument V. The resultant coordinates are passed back to the calling program through argument V.

The statement

\[ \text{DIMENSION V(3)} \]

is used to dimension a three-element array for the coordinates of the vector passed to this subroutine.

The statements

```plaintext
C COMPUTE UNIT VECTOR (DIRECTION COSINES OF VECTOR)
C
TEMP = SQRT(DOT(V,V))
V(1) = V(1) / TEMP
V(2) = V(2) / TEMP
V(3) = V(3) / TEMP
RETURN
```

are used to compute the scalar length of the vector by computing the square root of the vector dotted on itself. The length of each coordinate vector is divided by the scalar length of the vector, resulting in the direction cosines of the vector. These are returned to the calling program through argument V. The unit vector of a given vector \( \mathbf{A} \) can be expressed mathematically as

\[
\mathbf{A}_i = \frac{\mathbf{A}_i}{\sqrt{\mathbf{A} \cdot \mathbf{A}}}
\]

where \( i \) is the x, y, or z coordinate.
Function XDIST(XA, XB)

Function XDIST is called by Subroutines TRACK, CALC, TESTG, VOLUM, and DCOSP to compute the distance between two points passed into this routine through arguments XA and XB. This function subprogram computes the distance by utilizing the standard distance formula

\[ \text{Distance} = \sqrt{\sum (X_A - X_B)^2} \]

The statements

```c
C
C1 COMPUTE THE DISTANCE BETWEEN TWO GIVEN POINTS XA AND XB
C
DIMENSION XA(3), XB(3)
XSUM=0.
```

are used to dimension two three-element arrays for the coordinates of the two points passed into this routine. Variable `XSUM` is then initialized to zero.

The statements

```
DO 10 I=1,3
   XSUM=XSUM+(XA(I)-XB(I))**2
10 CONTINUE
```

consist of a DO loop which is used to sum the distance squared for each of the coordinates between the two points.

The statements

```c
XDIST=SQRT(XSUM)
RETURN
```

are used to compute the square root of the summed distance squared which results in the distance between the two given points. The routine returns to the calling program.
Subroutine DCOSP(XA, XB, WA)

Subroutine DCOSP is called by Subroutines CALC, TESTG, and VOLUM to compute the direction cosines from point XA to point XB. This is accomplished by calling Function XDIST to compute the scalar distance from point XA to XB; and dividing each coordinate vector between the two points by the distance between the two points.

The statement

```
COMPUTE THE DIRECTION COSINES FROM POINT XA TO POINT XB
AND STORE DIRECTION COSINES IN WA
```

```
DIMENSION XA(3), XB(3), WA(3)
```

is used to dimension three three-element arrays for the coordinates of the two points and the resultant direction cosines.

The statement

```
DIS=XDIST(XA,XB)
```

is used to call Function XDIST to compute the distance between the two given points passed to this subroutine.

The statements

```
DO 10 I=1,3
    WA(I)=(XB(I)-XA(I))/DIS
10 CONTINUE
RETURN
```

consist of a DO loop which is used to divide each coordinate vector between the two points by the distance between the two points. The result is the direction cosines of the vector from point XA to point XB. In mathematical terms

\[
\bar{WA}_i = \frac{XB_i - XA_i}{\sqrt{\sum(XA_i - XB_i)^2}}
\]

where \(i\) represents the x, y, or z coordinate of the point.
Subroutine TROPIC(WP)

Subroutine TROPIC is called by Subroutines GRID and AREA to calculate isotropic direction cosines. Each call to Subroutine TROPIC generates a different set of direction cosines.

The statement

```
  C  GENERATE RANDOM DIRECTION COSINES FROM AN ISOTROPIC DISTRIBUTION
  C  DIMENSION WP(3)
```

is used to dimension a three-element array for the coordinates of the generated random direction cosines.

The statements

```
10  X1=RAI (-1)  
  X2=RAI (-1)  
  X1=XS .X2
  X2=S .X2
  T=X1*XS
  IF(T.GE.1.)GOTO 10
```

are used to call function RAN to generate random numbers X1 and X2 and to compute the sum of the squares. A test is made to determine if $X1^2 + X2^2 \leq 1.0$. If not, the statements are repeated until the expression is satisfied.

The statements

```
  C  COMPUTE THE SINE AND COSINE OF A RANDOM ANGLE PHI
  C  C5PHI=(X1*X2)/T
  C  SINPHI=(2.*X1*X2)/T
```

are used to compute the cosine of random angle $\phi$ from the expression

$$ \cos \phi = \frac{X1^2 - X2^2}{X1^2 + X2^2} $$

and to compute the sine of random angle $\phi$ from the expression
\[
\sin \phi = \frac{2.0 \cdot x_1 \cdot x_2}{x_1^2 + x_2^2}
\]

The statements

\begin{verbatim}
X1=RAN (-1)
IF(X1*LE.*5) SNPHI=-SNPHI
\end{verbatim}

are used to call Function RAN to compute a random number between zero and one and to set \( \sin \phi \) negative if the random number was less than or equal to 0.5.

The statements

\begin{verbatim}
C4 COMPUTE THE SINE AND COSINE OF A RANDOM ANGLE THETA
C
CSTHT=2.*RAN (-1)-1,
SNHT SQRT(1.-CSTHT**2)
\end{verbatim}

are used to call Function RAN to compute a random number between zero and one, double it, and subtract one to arrive at the cosine of random angle \( \theta \). The sine of \( \theta \) is computed from the expression

\[
\sin \theta = \sqrt{1 - (\cos \theta)^2}
\]

The statements

\begin{verbatim}
C4 COMPUTE RANDOM DIRECTION COSINES
C
WP(1)=SNHT*SNPHI
WP(2)=SNHT*CSPHI
WP(3)=CSTHT
RETURN
\end{verbatim}

are used to compute the \( x \), \( y \), and \( z \) coordinates of the random direction cosines and store them in three-element array WP via the following expressions:

\[
\begin{align*}
WP(1) &= \sin \theta \cdot \sin \phi \\
WP(2) &= \sin \theta \cdot \cos \phi \\
WP(3) &= \cos \theta
\end{align*}
\]

the program returns to the calling program with WP.
Function \( S(I,N) \)

Function \( S \) is called by Subroutines RPPIN, RPP2, and RPP for the purpose of retrieving the coordinate of any one of the six sides of a rectangular parallelepiped (RPP). Given argument \( I \), the ordinal number of the RPP, and argument \( N \), the side number of the RPP where \( N = 1, 2, 3, 4, 5, \) or \( 6 \) (\( X_{\min}, X_{\max}, Y_{\min}, Y_{\max}, Z_{\min}, \) or \( Z_{\max} \) respectively), the routine will compute the location in the ASTER array of the required coordinate and return with the value of the coordinate set equal to \( S \).

The statements

```plaintext
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/GEOM/LBASE,RIN,ROUT,LR1,LRO,PINF,IER+DIS!
EQUIVALENCE(MASTER,ASTER)
```

are used to dimension the MASTER array for 30,000 words, to make the data in the ASTER array available to this routine, to pass the value of pointer LBASE into this routine, and to set the MASTER array equivalent to the ASTER array.

The statements

```plaintext
C S RETRIEVES COORDINATE OF ANY ONE OF THE 6 SIDES OF AN RPP
C I IS THE RPP NUMBER     N IS THE SURFACE NUMBER
L=LBASE+12*I-1+2*(N-1)
LL=MASTER(L+1)
S=ASTER(LL)
RETURN
```

are used to compute the location of the pointer data for the given side of the given RPP and to retrieve the pointer from the next word since that is the location of the pointer to the coordinate for the given side. Using this pointer, the coordinate is retrieved from the ASTER array and equated to \( S \). The routine returns to the calling program.
Subroutine RPP2(LSURF, XP, IRP)

Subroutine RPP2 is called by Subroutine Gl whenever an exit intersect occurs with a surface of the RPP containing the target geometry. The purpose of the subroutine is to determine the number of the RPP (if any) that the ray is entering upon leaving the present RPP. If no abutting RPP can be found, the routine returns a zero as the number of the abutting RPP.

The statement

```c
EJ
C
FINO NUMBER OF ABUTTING RPP TO INTERSECTED SURFACE
DIMENSION XP(3)
```

is used to dimension a three-element array to contain the coordinates of the exit intersect point of the present RPP.

The statements

```c
COMMON ASTER(10000)
COMMON/PAREM/XB(3)*WB(3)*IR
COMMON/GEOM/LBASE*RIN*ROUT+LRI+LRO+PINF+IERR+DIST
COMMON/UNCSEM/NTRIP+NSCAL+NBODY+NRMAX+LTRIP+LSCALE+LREOD:
1 LDATA+LRIN+LROT+L10+LOCOA+11S+130+LBODY+NASC+KLOOP
```

are used to make the contents of the ASTER array available to this routine and to pass the RPP number and the beginning location of the RPP data into this routine.

The statements

```c
C
LOC=LBASE+12*(NASC-1)-2*(LSURF+1)
CALL UN2(LOC+LOCAT+NC)
```

are used to compute the location of the pointer data for the surface number of the RPP that the ray is exiting; and to unpack the number of abutting RPP's to that surface and the pointer to the location in the MASTER array of the list of abutting RPP's.

The statements

```c
IF(NC=1) 10*20*30
```
are used to determine if there are no abutting RPP's, only one abutting RPP, or more than one abutting RPP to the intersected surface of the RPP. If there are no abutting RPP's, argument IRP is set to zero, and control is returned to Subroutine Gl.

The statements

\[ \text{10 } \text{IRP}=0 \]
\[ \text{RETURN} \]

are executed if there is only one abutting RPP to the intersected surface. These statements therefore unpack the abutting RPP from its location in the abutting RPP section and return control to Subroutine Gl with argument IRP set to the number of the abutting RPP.

The statement

\[ \text{20 CALL UN2(LOCAT+IRP+DUM)} \]
\[ \text{RETURN} \]

is executed if there is more than one abutting RPP to the intersected surface. Variable M is set to one and is subsequently used to determine which of two abutting RPP's in a packed word is to be tested.

The statements

\[ \text{30 } \text{M}=1 \]

are used to begin a DO loop which will test each abutting RPP to the intersected surface to determine which abutting RPP the ray is entering. Variable M is set to a negative one whenever DO loop index I is odd; M is set to a positive one when index I is even. For negative M, the left abutting RPP in the packed word is tested. For positive M, the right abutting RPP in the packed word is tested.

The statement

\[ \text{IF(M.GT.0)GOTO 50} \]
is used to test variable $M$ to determine which RPP in the packed abutting RPP word is to be tested.

The statements

```
CALL UN2(LOCAT*I1*I2)
LOCAT=LOCAT*I
IRP=I
GOTO 70
```

executed if $M$ is negative, are used to retrieve the next packed abutting RPP word. The left abutting RPP in the packed word is equated to argument $IRP$, and the program branches to determine if the RPP is an abutting RPP. LOCAT is indexed to the next abutting RPP word for unpacking when needed.

The statement

```
50 IRP=I2
```

executed when $M$ is positive, is used to set the right abutting RPP in the packed word equal to argument $IRP$ for subsequent testing.

The statement

```
70 LS=(1-LSURF)/2
```

is used to compute a control variable based on the intersected surface number in order to insure that identically numbered sides of the intersected RPP and the potential abutting RPP are never compared since these boundaries would never be together.

The statements

```
DO 80 J=1,3
IF(J.EQ.LS) GOTO 80
IF((S(IRP,2*J-1)=XP(J))*(XP(J)=S(IRP,2*J-1),LT,0)) GOTO 90
80 CONTINUE
RETURN
```
consist of a DO loop which is used to determine if the intersect occurs within the boundaries of the potential abutting RPP by evaluating, assuming an intersect on an X plane,

\[
Y_{\text{min}} < Y_c < Y_{\text{max}} \\
Z_{\text{min}} < Z_c < Z_{\text{max}}
\]

where \(Y_c\) and \(Z_c\) are the y-z coordinates of the intersect and \(Y_{\text{min}}, Y_{\text{max}}, Z_{\text{min}},\) and \(Z_{\text{max}}\) are the bounding plane coordinates of the potential abutting RPP.

The statements

```fortran
90 CONTINUE
   IRP=0
   RETURN
END
```

are used to continue to the next potential abutting RPP if the intersect did not occur within the boundaries of the present potential RPP. If the intersect occurs on the face of an RPP that has no abutting RPP, IRP is set to zero, and control is returned to Subroutine G1.
Subroutine VOLUM

Subroutine VOLUM is called by the MAIN program if the option variable for calling this subroutine is set to one. The purpose of this routine is to compute the volumes of each region within a given imaginary box, where the box is defined as in Figure 73.

N1 number of vertical cells
N2 number of horizontal cells
XV lower right corner of box front
XT upper right corner of box front
XO lower left corner of box front
XA lower right corner of box end
DT vertical dimension of cell
DOD horizontal dimension of cell
XVDIS distance from front to back of box

FIG. 73. Volume Geometry

Rays are traced from the lower right corner of each cell from the front to the back of the box and the distances through each region are stored in an array. When all rays have been traced and the total distance through each region accumulated, the volumes are computed from the region distances and the cell dimensions.
The statement

```c
COMPUTE VOLUMES BY REGION IN VOLUME DEFINED BY BOX
```

```c
DIMENSION VASTER(1000), WAB(3), WTB(3), WOB(3), DSP(3),
1 XV(3), XT(3), XA(3), XO(3), XP(3), XTEMP(3)
```

is used to dimension arrays for use by this subroutine.

The statements

```c
COMMON ASTER(1000)
COMMON/PAREM/XB(3), WB(3), IR
COMMON/GEOM/LBASE, RIN, ROUT, LRI, LRO, PIN, IERR, DIST
COMMON/NC5BEM/NRPP, NTRIP, NSCAL, NBODY, NMAX, LTRIP, LSCALE, LREGO,
1 LOATA, LRI, LROT, LO, LOCDA, I15, I30, LBODY, NASC, KLOOP
COMMON/WALT/LIRFO, N81ERR
```

are used to pass information into and out of this subroutine.

The statements

```c
901 FORMAT(3E20.8)
902 FORMAT(2E20.8)
903 FORMAT(1M0, 10X*6HVERTEX, 14X*6HTOP, PT, 14X*6HBOT, PT, 14X*7HSIDE, PT)
904 FORMAT(4E20.8)
905 FORMAT(1M0, 8X*12HDELTA ON TOP, E20.8, 10X*10HSIDE DELTA, E20.8)
906 FORMAT(2I10)
908 FORMAT(1M0, 2X*18HSTARTING REGION I5, I5)
909 FORMAT(1M0, 16HVASTER OVERWRITE, 5X*6HNRMAX, I5)
910 FORMAT(1I10, E20.8)
911 FORMAT(1M0, 6HBAD CARD/I10*E20.8, 1H NOT PROCESSED)
912 FORMAT(1I10, E20.8, 5X*5E20.8, 5X*E20.8)
913 FORMAT(1M0*5HSUMV=, 5X*E20.8)
```

are used to format data for input and output.

The statements

```c
READ (5, 906) IR, N81ERR
IF (N81ERR .LE. 0) N81ERR = 25
```

are used to enter the region number of starting point XV and to enter the allowable number of Subroutine Gl errors. If the number of allowable errors entered is less than or equal to zero, the allowable errors in Subroutine Gl are set to 25.
The statements

**ENTER COORDINATES OF BOX**

```fortran
READ (5,901) (XV(I),I=1,3)
READ (5,901) (XT(I),I=1,3)
READ (5,901) (XO(I),I=1,3)
READ (5,901) (XA(I),I=1,3)
```

**ENTER CELL SIZE**

```fortran
READ (6,902) DD0,DT
WRITE (6,903)
WRITE (6,904) (XV(J),XT(J),XO(J),XA(J),J=1,3)
WRITE (6,905) DD0,DT
WRITE (6,908) IR
```

are used to enter and print out the coordinates of the four corners that define the box; to enter and print out the cell dimensions; and to print out the starting region number.

The statement

```fortran
IF(NRMAX.GT.2000)WRITE (6,909)NRMAX
```

is used to determine if the number of regions used to describe the target geometry is excessive relative to the array size used in this subroutine. If it is, an error message with the number of regions is printed out.

The statements

```fortran
CALL DCOSP(XV,XT,WTB)
CALL DCOSP(XV,XO,WOB)
CALL DCOSP(XV,XA,WAB)
```

are used to compute the direction cosines of the vectors from the vertex point to each of the three other points of the box.
The statement

\[ XVDIS = \text{XDIST}(XV,XA) \]

is used to compute the length of the box from front to back by the use of Function XDIST.

The statements

\begin{verbatim}
TESTON=0.
TESTOV=0.
XTEMP(1)=0.
DO 10 J=1,NRMAX
  VASTER(I)=0.
10 CONTINUE
\end{verbatim}

are used to zero subroutine variables and the array for storing the distances through the regions.

The statements

\begin{verbatim}
JRJIR
JRJIR
\end{verbatim}

are used to save the starting region number for later use in the subroutine.

The statements

\begin{verbatim}
C C
C
C COMPUTE NUMBER OF HORIZONTAL AND VERTICAL CELLS
C
N2=\text{XDIST}(XV,XO)/DOD+1.
N1=\text{XDIST}(XV,XT)/DT+1.
\end{verbatim}

are used to compute the number of horizontal and vertical cells from the dimensions of the cell and the size of the front plane of the box.
The statement

```
C6  TRACE RAYS FROM LOWER RIGHT CORNER OF EACH CELL
   DO 300 J=1,N2
```

is used to begin a DO loop which will trace a ray from the lower right corner of each cell to the end of the box, accumulating distances that the rays travel through each region. This DO statement indexes columns of cells on the face of the box.

The statements

```
DO 100 I=1,3
DSP(I)=WTB(I)*DT
X6(I)=X(V(I))
W6(I)=WAB(I)
100 CONTINUE
```

consist of a DO loop which is used to compute the vector from the present origin to the next origin in the current column from where the next ray is to be traced. The coordinates of the origin and the direction cosines of the current ray are then assigned.

The statements

```
S1=0.
IR=JIR
```

are used to initialize variable S1, the distance to the next region, and to initialize to the region number of the origin of the current ray.

The statements

```
C6  TRACE ALL RAYS FROM COLUMN OF CELLS
   DO 200 I=1,N1
      NASC=-1
```

are used to begin a DO loop for tracing each ray from each cell in the present column of cells. For each new ray, NASC is set to a -1 to indicate to Subroutine GI that a new ray is to be traced.
The statements

```
    TRACE RAY THROUGH BOX VIA SUBROUTINE G1

110 CALL Gl(S1,IRPRIM,XP)
   IF (IERR.GE.NG) ERR GOTO 400
   VASTER(IR)=VASTER(IR)+S
   IF (DIST.GE.XVDIS) GOTO 115
   IF (IRPRIM.LE.0) GOTO 120
   IR=IRPRIM
   GOTO 110
```

are used to call Subroutine Gl to trace the given ray to the end of the box. Subroutine Gl returns when a new region has been encountered and the number of errors that occurred in Subroutine Gl is compared to the allowable limit. If the allowable number of errors was exceeded, the program is terminated. If the allowable error limit was not exceeded, the distance through the given region is added to the previous distance that the ray(s) has travelled through the region. A test is made to determine if the new region was encountered beyond the end of the box. If so, the routine branches to subtract that distance beyond the end of the box. A test is made to determine if the new region is an RPP boundary; if so, the ray is terminated. If the new region was encountered before the end of the box, the present region is updated to the new region, and the program branches to call Subroutine Gl again to continue the ray.

The statement

```
   115 VASTER(IR)=VASTER(IR)-(DIST-XVDIS)
```

which is executed when the new region encountered lies beyond the end of the box, is used to subtract that portion of the distance beyond the end of the box.

The statements

```
120 XTEMP(1)=WB(1)
   XTEMP(2)=WB(2)
   XTEMP(3)=WB(3)
   IR=IR
```

are used to save the ray direction cosines and to initialize to the starting region number of the present ray.
The statements

\[
\text{TSTON=TSTON+DT} \\
\text{IF (TSTON.GT.0.) GOTO 180}
\]

are used to determine if the origin of the next ray in the current column is in the same region as the origin of the ray just completed.

The statements

\[
\begin{align*}
\text{WB(1)=WTB(1)} \\
\text{WB(2)=WTB(2)} \\
\text{WB(3)=WTB(3)} \\
\text{NASC=1}
\end{align*}
\]

are used to assign direction cosines of the vector from point \(X_V\) to \(X_T\), and to initialize for a new ray to determine the region number of the next origin.

The statements

\[
\text{CALL G1(S1,IRPRIM,XP)} \\
\text{IF (IERR+GE.+NG1ERR) GOTO 400} \\
\text{IF (S1=DT) 130+160+170}
\]

are used to call Subroutine G1 to find the distance to the next region in the direction of the next origin. If an excessive number of errors did not occur in Subroutine G1, the distance to the next region is compared with the distance to the next origin.

The statements

\[
\begin{align*}
130 & \text{ IR=IRPRIM} \\
& \text{JIR=IR} \\
& \text{CALL G1(S1,IRPRIM,XP)} \\
& \text{IF (IERR+GE.+NG1ERR) GOTO 400} \\
& \text{IF (DIST=DT) 140+160+170}
\end{align*}
\]
are executed if the distance to the next origin is greater than the distance to the next region. These statements update to the new region number and save it in case the next origin is in that region. If an excessive number of errors did not occur in Subroutine Gl, the distance to the next region is compared with the distance to the next origin to determine if the origin is in the new region returned by Subroutine Gl.

The statements

140 IF(IRPRIM)150+210+130
150 STOP

are executed if the distance to the next origin is again greater than the distance to the next region. If this next region number is negative, the program stops because this is an error. If the region number is zero, the RPP enclosing the target geometry has been intersected and the program branches to index to the next column of cells. If the region number is positive, the program branches to find the next region where the origin could be located.

The statements

160 IR=IRPRIM
    JIR=IR

are executed if the next origin and a new region occur simultaneously. These statements update to the new region and store it in a save area for later use.

The statement

170 TESTON=51

which is executed if the distance to the next region is greater than the distance to the next origin, is used in the program to determine if succeeding ray origins are in the same region.
The statements

C 180 DO 190 JI=1,3
   WB(JI)=XTEMP(JI)
   XB(JI)=XB(JI)+DSP(JI)
190 CONTINUE
200 CONTINUE

consist of a DO loop which is used to compute the coordinates of the origin of the next ray and to restore the original direction cosines of the ray. The program then loops to trace the next ray.

The statement

C 210 NASC=-1

is used to set NASC to a -1 to indicate to Subroutine G1 that a new ray is to be traced. This statement is executed when all of the rays in one column have been traced, and the rays in the next column are now to be traced.

The statements

DO 220 I=1,3
   WB(I)=W0B(I)
   XB(I)=XB(I)
220 CONTINUE

consist of a DO loop which is used to assign the direction cosines and origin of a ray to be traced from the first origin of the current column to the first origin of the next column.
The statements

```
JIR=IRJ
IR=JIR
TESTDN=0
TESTOV=TESTOV+DOD
IF (TESTOV) 230,230,260
```

are used to initialize to the region number of the first origin of the current column. TESTDN is initialized to zero for later use in determining the regions of the origins of the next column. The distance to the next region in the direction of the first origin of the next column is computed to determine if the first origin of the next column is in the same region as the first origin of the current column.

The statements

```
C12  DETERMINE REGION OF FIRST ORIGIN OF NEXT COLUMN
C
  230 CALL 01(S1,IRPRIM,XP)
  IF (IERR.GE.NG) IERR GOTO 400
  IF (S1.DOD) 240,260,270
```

are executed if the region number of the first origin of the next column is not known. These statements are used to compute the distance to the next region in the direction of the first origin of the next column by calling Subroutine G1. This distance is compared with the distance between the columns after checking for more-than-allowable errors.

The statements

```
240 IR=IRPRIM
IR=IR
  CALL 01(S1,IRPRIM,XP)
  IF (IERR.GE.NG) IERR GOTO 400
  IF (DIST.DOD) 250,260,270
```

are executed if the distance to the first origin of the next column is greater than the distance to the next region. These statements update to the new region number and save it in case the origin is in this region. Subroutine G1 is again called to determine the distance to the following region; Subroutine G1 compares the total distance with the distance to the origin, after checking for more-than-allowable errors.
The statements

250 IF(IRPRIM).EQ.255.400.230
255 STOP

are executed if the distance to the first origin of the next column is again greater than the distance to the next region. If this next region number is negative, the program stops since this is an error. If the region number is zero, the RPP enclosing the target geometry has been intersected, and the program branches to index to the next column. If the region number is positive, the program branches to find another region where the next origin could be located.

The statements

260 IR=IRPRIM
1RJEIR

are executed if the first origin of the next column and an intersect with a new region occur simultaneously. These statements update to the new region and save it for later use.

The statement

270 TESTO=SI

which is executed if the distance to the next region is greater than the distance to the next origin, is used in the program to determine if succeeding ray origins are in the same region as previous ray origins.

The statements

C 13 SHIFT ORIGIN OF NEXT RAY TO FIRST ORIGIN OF NEXT COLUMN OF CELLS
C
280 DO 290 I=1,3
XA(I)=XA(I)+WOB(I)*ODD
XV(I)=XV(I)+WOB(I)*ODD
XT(I)=XT(I)+WOB(I)*ODD
290 CONTINUE

consist of a DO loop which is used to shift the vertex, top point and the box end point to the next column of cells.
The statements

JIR=IR
300 CONTINUE

are used to save the region number of the first origin of the new column
and to branch to trace the rays of the new column.

The statements

C C14  ALL RAY DISTANCES THROUGH EACH REGION IN BOX ACCUMULATED
C 400 READ (5,910)IR1,VR
   IF(IFE.R.GE.NG)ERR1GOTO 500
   IF(IR1.LE.0)IR1=NRMAX+1
   SUMV=0.

are executed when all rays have been traced; when more than the allowable
limit of errors occurred in Subroutine Gl; or when the front plane of the
box intersects an RPP boundary. These statements are used to enter a
region number and its pre-computed volume. The Subroutine Gl error
counter is compared with the allowable limit of errors; if more than the
allowable limit of errors occurred, the program branches to return control
to the MAIN program. If less than the number of allowable errors occurred
in Subroutine Gl, and no region number was entered, the region number is
equated to one greater than the number of regions in the target geometry.
SUMV, the variable for summing the total volume of all regions within the
box, is also set to zero.

The statement

C C15  COMPUTE VOLUME OF EACH REGION IN BOX
C DO 450 I=1,NRMAX

is used to begin a DO loop which will compute and print out the volume
of each region encountered within the volume of the box and to compute
the percent error with the pre-computed volumes if the pre-computed
volume was entered.
The statement

\[ \text{VASTER(I)} = \text{VASTER(I)} \times \text{DOODOT} \]

is used to compute the volume of each region within the volume of the box.

The statement

\[ \text{IF}(\text{I=IR1}) \]

\[ 410 \times 430 \times 420 \]

is used to determine if the current region of the DO loop has a pre-computed volume entered for computing percent error.

The statements

\[ 410 \text{ WRITE (6,910)I,VASTER(I)} \]

\[ \text{GOTO 440} \]

are used to write out the region number and the computed volume of the region if the entered region number is greater than the current region number of the DO loop.

The statements

\[ 420 \text{ WRITE (6,911)IR1,VR} \]

\[ \text{READ (5,910)IR1,VR} \]

\[ \text{GOTO 410} \]

are executed if the region number of the DO loop is greater than the last region number entered. These statements are used to print out the entered region number and pre-computed volume and to enter the next card with region number and pre-computed volume.

The statements

\[ \text{C 430 XPERC=100*#(VASTER(I))/VR-1,7} \]

\[ \text{WRITE (6,912)I,VASTER(I),VR,XPERC} \]
are executed when the entered region is equal to the region number of the DO loop. These statements are used to compute the percent error between the computed volume and the pre-computed volume of the current region. The region number, computed volume, pre-computed volume, and percent error are printed, and the next region number with pre-computed volume is entered.

The statements

```
VASTER(I)=VR
READ (5,910)IR1,VR
```

are used to accumulate the total volume of the regions considered and to transfer to consider the next region.

The statements

```
   440 SUMV=SUMV+VASTER(I)
   450 CONTINUE
```

are used to print out the total of the volume of all of the regions encountered within the box. The Subroutine G1 error counter is set to zero, and this subroutine returns control to the MAIN program.
Subroutine AREA

Subroutine AREA is called by the MAIN program if the option variable for calling this subroutine is set to one. The purpose of this subroutine is to compute the presented area of an object or target as viewed from any azimuth and elevation angle.

The statement

\[ \text{DIMENSION XP(3), WP(3), XB(3), CONVRT(15), TYPEUN(4)} \]

is used to dimension arrays for use in this subroutine.

The statements

\[ \text{COMMON ASTER(10000)} \]
\[ \text{COMMON/PAREM/XP(3), WP(3), IR} \]
\[ \text{COMMON/GEOM/LBASE, RIN, ROUT, LRI, LRO, PINF, IERR, DIST} \]
\[ \text{COMMON/UNCDEM/NRP, NTRIP, NSCAL, NBOY, NMAX, LTRIP, LSCAL, LREDG,} \]
\[ \text{LDATA, LRIN, LROT, L10, LOCA, ID, LBOY, NASC, KLOOP} \]
\[ \text{COMMON/CAL/NIR, SLO, ANGLE, NTY, SSPACE, LXS(3), WS(3),} \]
\[ \text{TRAVEL, SN, V, H, IVIM} \]
\[ \text{COMMON/WALT/LIRFO, NG, IERR} \]
\[ \text{COMMON/CAL/CELL/CESIZ} \]
\[ \text{COMMON/ENG/LE0} \]

are used to pass information into and out of this subroutine.

The statements

\[ \text{C} \]
\[ \text{FORMAT(1I10, 6X, 2A2)} \]
\[ \text{FORMAT(6F12.5)} \]
\[ \text{FORMAT(1HO, 22HMEMORY OVERLAP IN AREA, 5X, 7HGEOM = I6,} \]
\[ \text{5X, 6HAREA = I6, 5X, 6HDIRFO = I6)} \]
\[ \text{FORMAT(1HO, 13HERROR IN AREA, 5X, 9HICODE = 0)} \]
\[ \text{FORMAT(1HO, 8HAZIMUM = F10.3, 5X, 10HLEVATION = F10.3,} \]
\[ \text{FORMAT(1HO, 12HCELL SIZE = F6.1, 1X, 1HA, F6.1, 1X, 1A2,} \]
\[ \text{1H = 12HAREAS IN SQ, 1X, 4A2, 1H0,} \]
\[ \text{FORMAT(1HO, 5HICODE = 19X, 6AREA, 1F12.5)} \]
\[ \text{FORMAT(15X, 6F12.5, 1X, 15HPRESENTED AREA = F12.5,} \]
\[ \text{FORMAT(15X, 18HMNUMBER OF CELLS = 15+10X,} \]
\[ \text{12HNUMBER OF CELLS = 15+15)} \]

are used to format data for input and output and to format output messages.
The statements

```
DATA HMIN, HNFT, HMCN, HNMB/2HMIN, 2HNFT, 2HMCN, 2HNMB /
TYPEUN(1) = HMIN
TYPEUN(2) = HNFT
TYPEUN(3) = HMCN
TYPEUN(4) = HNMB
```

are used to enter Hollerith constants data. These constants, which are codes for inches, feet, centimeters, and meters, are equated to the four elements of array TYPEUN.

The statements

```
CONVRT(1,1) = 1,
CONVRT(1,2) = 0.00694444444444,
CONVRT(1,3) = 0.0254, 625006
CONVRT(1,4) = 0.00694444444444,
CONVRT(2,1) = 144,
CONVRT(2,2) = 1,
CONVRT(2,3) = 929.0361161
CONVRT(2,4) = 0.9909341161
CONVRT(3,1) = 15499969
CONVRT(3,2) = 0.0107636736
CONVRT(3,3) = 1,
CONVRT(3,4) = 1
CONVRT(4,1) = 15499969
CONVRT(4,2) = 10.7636736
CONVRT(4,3) = 10000,
CONVRT(4,4) = 1,
```

are used to assign constants for converting the area from one unit of measurement to another. The constants for array CONVRT are used for converting an area as follows:

- CONVRT(1,1) square inches to square inches
- CONVRT(1,2) square inches to square feet
- CONVRT(1,3) square inches to square centimeters
- CONVRT(1,4) square inches to square meters
- CONVRT(2,1) square feet to square inches
- CONVRT(2,2) square feet to square feet
- CONVRT(2,3) square feet to square centimeters
- CONVRT(2,4) square feet to square meters
- CONVRT(3,1) square centimeters to square inches
- CONVRT(3,2) square centimeters to square feet
- CONVRT(3,3) square centimeters to square centimeters
- CONVRT(3,4) square centimeters to square meters
CONVRT(4,1) square meters to square inches
CONVRT(4,2) square meters to square feet
CONVRT(4,3) square meters to square centimeters
CONVRT(4,4) square meters to square meters

The statement

BLANK=WHBB

is used to equate two Hollerith blanks to variable BLANK for determining if there are blanks in card input fields.

The statements

```
C1 COMPUTE AND INITIALIZE AREA FOR STORING PRESENTED AREA
    BY COMPONENT CODE
C
LAREA=LIRFO=1000
IF(LAREA<GE.0E.0E0)GOTO 10
```

are used to compute a starting location in the ASTER array for storing the presented area of the target by component code (1-999). The location is computed to be 1000 words before location LIRFO, the storage area for the identification code and component code data. A test is made to verify that this new storage area does not overlap the end of the target geometry data.

The statements

```
WRITE (6,908)LEGEOM,LAREA,LIRFO
STOP
```

are executed if the new storage area overlaps the end of the target geometry data in the ASTER array. These statements cause an error message to be printed out with the pertinent subroutine variables and to terminate the program.

The statements

```
10 LAREA1=LIRFO
    DO 20 L=LAREA,LAREA1
        ASTER(L)=0.
    20 CONTINUE
```

are used to compute the last word of the array for storing the presented area of the target by component code and to zero the entire array with the DO loop.
The statements

```
C2  READ GRID INPUT PARAMETERS
   READ (5,901)NX,NY,1RSTRT,IERC,N1ERR,NSTRT,NEND,CELLUN,AREAUN
   READ (5,902)AREA,LENGTH,ZSHIFT,GROUND
   READ (5,902)XSHIFT,YSHIFT,CELSIZ
```

are used to enter the grid input parameters.

The statements

```
C3  INITIALIZE PARAMETERS NOT SET BY INPUT
   IF(1RSTRT.LE.0)IRSTRT=1
   IF(CELSIZ.LE.0)CELSIZ=4.
   IF(N1START.LE.0)NSTRT=1
   IF(N1ERR.LE.0)N1ERR=25
   IF(AREAUN.EQ.BLANK)AREAUN=MHIN
   IF(CELLUN.EQ.BLANK)CELLUN=MHIN
```

are used to determine or initialize the starting region number, the cell size, the cell number from which ray tracing is to begin, the limit of allowable errors that can occur in Subroutine G1, the measurement units for expressing the area, and the measurement units of the cell.

The statements

```
C4  DETERMINE MEASUREMENT UNITS AND COMPUTE GRID CELL AREA
   DO 30 I=1,4
      IF(CELLUN.EQ.TYPEUN(I))GOTO 40
      CONTINUE
   30 CONTINUE
   40 DO 50 J=1,4
      IF(AREAUN.EQ.TYPEUN(J))GOTO 60
      CONTINUE
   50 CONTINUE
   60 AREA=CELSIZ*CELSIZ*CONVRT(I,J)
```

are used to determine the measurement units of the cell and the desired measurement units to describe the area. The area of the cell is computed and converted to the selected measurement units (inches$^2$, feet$^2$, centimeters$^2$, or meters$^2$).
The statements

C

RADIAN=0.017453292519943
AR=A*RADIAN
ER=E*RADIAN
SA=SIN(AR)
CA=COS(AR)
SE=SIN(ER)
CE=COS(ER)

are used to compute the sine and cosine of the azimuth and elevation angles from the azimuth and elevation angles that were entered in degrees.

The statements

KL=NX*NY
NHIT=0

are used to compute the number of cells in the grid plane and to initialize the number of hits counter, NHITS, to zero.

The statement

C

PROCESS KL CELLS IN GRID PLANE
C

DO 200 KSTART,KL

is used to begin a DO loop which will construct a grid plane, fire a ray from each cell in the grid plane, and compute and store the presented area by component code when the ray intersects the target.

The statements

WB(1)=CE*CA
WB(2)=CE*SA
WB(3)=SE

are used to compute the direction cosines for the ray normal to the grid plane and directed toward the target geometry.
The statements

```c
   COMPUTE ROW AND COLUMN NUMBER OF GRID CELL
   
   II=(KK-1)/NX+1
   JJ=KK-(II-1)*NX
```

are used to compute the row and column of a specific grid square from which the ray is to be fired toward the target geometry.

The statements

```c
   CELL2=.5*CELSIZ
   V=FLOAT((NY/2)=II)*CELSIZ +CELL2
   VREF=V+CELL2
   H=FLOAT((NX/2)-J)*CELSIZ +CELL2
   HREF=H+CELL2
```

are used to locate the lower left corner of the grid square. V represents the vertical distance from the center of the grid plane and H represents the horizontal distance. VREF and HREF refer to the center of the current grid square.

The statements

```c
   IV=RAN(-1)*10.0
   IH=RAN(-1)*10.0
   IVH=10.0*IH+IV
```

are used to compute two random numbers between zero and nine.

The statements

```c
   COMPUTE RANDOM POINT WITHIN GRID CELL
   
   V=V+CELSIZ*FLOAT(IV)/10.+CELSIZ/20.
   H=H+CELSIZ*FLOAT(IH)/10.+CELSIZ/20.
```

are used to locate one random point out of a possible 100 random points within the current grid cell.
The statements

\begin{align*}
  &XBS(1) = XSHIFT \cdot WP(1) + 1 \cdot 0E-4 \\
  &XBS(2) = XSHIFT \cdot WP(2) + 1 \cdot 0E-4 \\
  &XBS(3) = XSHIFT \cdot WP(3) + 1 \cdot 0E-4
\end{align*}

are used to transform the point within the current grid cell to the coordinate system of the target and, at the same time, to effectively move the grid plane and target system coordinate origins to a new location specified by the variables XSHIFT, YSHIFT, and ZSHIFT.

The statement

\begin{verbatim}
CALL TROPIC(WP)
\end{verbatim}

is used to call Subroutine TROPIC which generates random direction cosines from an isotropic distribution.

The statements

\begin{align*}
  &XBS(1) = XBS(1) \cdot WP(1) + 1 \cdot 0E-4 \\
  &XBS(2) = XBS(2) \cdot WP(2) + 1 \cdot 0E-4 \\
  &XBS(3) = XBS(3) \cdot WP(3) + 1 \cdot 0E-4
\end{align*}

are used to move the point within the current grid cell by a very small amount in a random direction.

The statements

\begin{verbatim}
CONVAT GAJO PLAN£ COOA0JN£S TO COOA0IN£S 0, TAAG\[T
\end{verbatim}

\begin{align*}
  &XB(1) = XBS(1) \cdot ENGTH \cdot WB(1) \\
  &XB(2) = XBS(2) \cdot ENGTH \cdot WB(2) \\
  &XB(3) = XBS(3) \cdot ENGTH \cdot WB(3)
\end{align*}

are used to back the point out of the target by an amount ENGTH, which was entered with the input data. The ray will originate from this point and pass through the position of the point before it was moved to its present position.

The statement

\begin{verbatim}
IF(XB(3) \leq GROUND) GOTO 200
\end{verbatim}

is used to determine the position of the origin of the ray. If the origin occurs below ground level, the ray will not be fired, and the program branches to compute the origin of the next ray.
The statements

```c
C C9 TRACE RAY TO FIRST TARGET COMPONENT HIT
C IR=IRSTRT
NASC=-1
110 CALL Gl(S1,IRPRIM,XP)
```

are used to initialize the region identifier to the starting region of the ray and to initialize the variable NASC to -1 to indicate to Subroutine Gl that a new ray is being fired. Subroutine Gl is called to move the point on the ray to the next region.

The statements

```
IF(IERR.GE.NB1) RETURN
IF(IRPRIM.LT.0) GOTO 200
IF(NASC.LE.NRPP) IRPRIM=0
IF(IRPRIM.EQ.0) GOTO 200
```

are used to determine if more than the allowable number of errors occurred in Subroutine Gl. If not, further tests are made to determine if an RPP boundary has been intersected. If the intersect occurs at an RPP boundary, the ray has missed the target and the program branches to process the next ray.

The statements

```
LOC=LIRFO+IRPRIM-1
CALL UNZ2(LOC,ICODE,IDENT)
```

are used to locate and unpack the identification code and the component code from the ASTER array for the region returned by Subroutine Gl.

The statements

```
IDENT=IDENT-1
IF(IDENT=(IDENT/10)*10,EQ,0) GOTO 120
IR=IRPRIM
GOTO 110
```

are used to determine if the material of the intersected region is part of the target. If not, the region number is updated, and control is returned to again call Subroutine Gl to return the next region encountered along the ray. One is subtracted from variable IDENT, since a one was added before IDENT was packed to prevent packing a negative number.
The statements

120 IF(ICODE.NE.0) GOTO 130
  WRITE (6,909)
  GOTO 200

are used to test the component code if the identification code test revealed
that the target was intersected. If the component code is zero, an error
message is printed out, and the program continues to process the next ray.

The statements

130 LOC=AREA+ICODE=1
  ASTER(LOC)=ASTER(LOC)+AREA
  NHIT=NHIT+1
  CONTINUE

are executed if both the identification code and component code agree that
the target has been hit by the current ray. These statements compute a
storage location in the ASTER array and add the presented area of the target
indexed by the component code of the material hit. The number of hits
counter is incremented by one for the current ray, and the program branches
to process the next ray.

The statements

C 9  PRINT RESULTS
  WRITE (6,910) A,E
  WRITE (6,911) CELSIZ, CELSIZ, CELLUN,AREAUN
  WRITE (6,912)
  SUMA=0.

are executed when all of the rays of the grid have been processed. These
statements are used to print out the azimuth and elevation angles of the
grid plane, the dimensions of the cells, and the measurement units of the
cell and of the computed areas. Column headings for printout of the pre­
sented areas by component codes are printed out, and the storage location
for summing all of the presented area is initialized to zero.
The statements

DO 250 I=1,999
   LOC=AREA*I=1
   IF(ASTER(LOC).EQ.0)GOTO 250
   WRITE (6,913)I,ASTER(LOC)
   SUMA=SUMA+ASTER(LOC)
250 CONTINUE

consist of a DO loop which is used to index each of the elements in the array for storing presented areas by component code (1-999). If there is no presented area for a given component code, the loop indexes to the next component code location. If there is a presented area for a given component code, the component code and its presented area are printed out. The presented area is added to location SUMA to obtain a total presented area.

The statements

WRITE (6,914)SUMA
WRITE (6,915)KLNHIT
RETURN
END

are executed when all of the presented areas by component code have been printed out and all of the presented areas have been summed. These statements print out the total presented area, the number of rays, and the number of rays that hit the target; and return control to the MAIN program.
Subroutine TESTG

Subroutine TESTG is called by the MAIN program if the option variable for calling Subroutine TESTG is set to one. The purpose of this routine is to trace a given number of rays between different sets of two points when given the coordinates and the region number of the points. Outputs of this routine consist primarily of leaving region, entering region, distance the ray has travelled into a region, coordinates of the point on the ray, and total distance the ray has travelled.

The statement

```c
TRACE A RAY BETWEEN TWO GIVEN POINTS XB TO XBF
```
```c
DIMENSION XP(3),XB(3)
```

is used to dimension two three-element arrays for the coordinates of the beginning and ending points.

The statements

```c
COMMON/PAREM/XB(3),WB(3),IR
COMMON/GEOM/LBASE,RIN,ROUT,LRI,LRO,FINF,IERR,DIST
COMMON/UNCGEM/NRPP,NTRIP,NSCAL,NBODY,NMAX,LTRIP,LSCALE,LREGO
1 LDATA,LRIN,LROUT,LLO,LDCDA,D15,D30,LBODY,NASC,KLOOP
COMMON/WALT/LIRF0,NG1ERR
```

are used to pass information into and out of this subroutine via COMMON statements.

The statements

```c
901 FORMAT(2110)
902 FORMAT(1H0+22HNUMBER OF SPECIAL RAYS,I5)
903 FORMAT(3E15.7,315)
904 FORMAT(1H0+5HSTART=5X,4HXB=3E15,7.8H IRSTRT=15/
1 14H END=7X,4HXB=3E15,7.8H IEND=15)
905 FORMAT(1H0+3HHB=3E15,7.8X,6MRANGE=8E15,7)
906 FORMAT(1H0+8X,2HIFLX=13X,2HIFLX=13X,2HIFLX=13X,2HYP, 
1 113X,2HYP+13X,4HOUT)
907 FORMAT(2110+5X,5E15,7)
908 FORMAT(1H0+21HTROUBLE IN REGION IR=I10)
```

are used to format data for input and output and to format output messages.
The statements

```c
C C ENTER NUMBER OF RAYS
READ (5,901)NRAYS,NG1ERR
WRITE (6,902)NRAYS
IF(NG1ERR.LE.0)NG1ERR=25
```

are used to enter and print out the number of different rays to be traced and to enter the number of allowable errors for Subroutine G1. If the number of allowable G1 errors entered is equal to or less than zero, it is set to 25 allowable Subroutine G1 errors.

The statements

```c
C C TRACE GIVEN NUMBER OF RAYS
DO 50 IRAY=1,NRAYS
```

are used to begin a DO loop which will enter and print out the data for each ray to be traced between two given points.

The statements

```c
C C ENTER POINT COORDINATES AND REGION OF EACH
READ (5,903)XB,IRSTR
READ (5,903)XB,IRFIN
WRITE (6,904)XB,IRSTR,IRF
```

are used to enter and print out the coordinates of the beginning and ending points of the ray and the region number where each point is located.

The statements

```c
RANGE=XDIST(XB,IRF)
CALL DCOSP(XB,IRF,WB)
WRITE (6,905)WB,RANGE
```

are used to compute the distance between the two points using Function XDIST, to compute the direction cosines of the line between the two points using Subroutine DCOSP, and to print out the distance and the direction of the line between the two points.
The statements

IR=IRSTRT
NASC=-1
WRITE (6,906)

are used to initialize variable IR to the starting region number and to set variable NASC to -1 to indicate to Subroutine G1 that a new ray is to be fired. The table headings for the output are then printed out.

The statements

C CTRACE RAY TO NEXT REGION INTERSECT
C 10 CALL G1(S1,IRPRIM,XP)
   IF(IERR.GE.NG1ERR)GOTO 60
   WRITE (6,907)IR,IRPRIM,S1,XP,DIST

are used to call Subroutine G1 to determine the distance to the next region, the number of the next region, the coordinates at the intersect of the next region, and the total distance the current ray has travelled from the first point. If more than the allowable number of errors occurred in Subroutine G1, control is transferred to zero the error counter and return to the MAIN program. If less than the allowable number of errors occurred in Subroutine G1, the present region number, the entering region number, the distance to the entering region, the coordinates of the intersect at the new region, and the total distance from the first point to the new region are printed out under the applicable column headings for this intersect.

The statements

IF(DIST.GE.RANGE)GOTO 30
IF(IRPRIM.LE.0)GOTO 20
IR=IRPRIM
GOTO 10

are used to determine if the distance that the ray has travelled is greater than or equal to the distance between the two points, or if the new region is outside the enclosing geometry of the enclosing RPP. If not, the current region number is updated to the new region, and control is branched to again call Subroutine G1 to continue the ray.
The statements

```c
20 WRITE (6,908) IR
GOTO 50
```

are executed if the new region is outside the enclosing RPP of the target geometry. These statements print out the old region number with an error message and then branch to start the next ray (if any).

The statements

```c
30 IF(IR.NE.IRFIN)GOTO 20
50 CONTINUE
```

are executed if the distance travelled is greater than or equal to the distance between the two points. These statements determine if the old region is the same as the region of the end point. If not, the old region number, with an error message, is printed out, and the program continues with the next ray (if any). If the regions are the same, the program continues with the next ray (if any).

The statements

```c
60 IERR=0 RETURN
END
```

are executed when all rays have been processed or when the allowable number of errors in Subroutine G1 was exceeded during the processing of one of the rays. These statements set the error counter to zero and return control to the MAIN program.
LIST OF SYMBOLS AND ABBREVIATIONS (SIMULATION MODEL)

Definitions of variable names utilized in this analysis program are contained in the following list of symbols and abbreviations (simulation model).

The variable names are presented in the following groups:

1. COMMON Statements

   Variable names appearing in the COMMON statements of all routines are listed in one group.

2. SUBROUTINES

   Variable names not appearing in COMMON statements are listed by subroutine.
# LIST OF SYMBOLS AND ABBREVIATIONS
## (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGLE</td>
<td>α</td>
<td>Angle between the normal and the ray at the intersect</td>
<td>Degrees</td>
</tr>
<tr>
<td>CA</td>
<td>cos α</td>
<td>Cosine of the azimuth angle of the ray with respect to the target geometry origin</td>
<td>ND</td>
</tr>
<tr>
<td>CE</td>
<td>cos θ</td>
<td>Cosine of the elevation angle of the ray with respect to the target geometry origin</td>
<td>ND</td>
</tr>
<tr>
<td>CELSIZ</td>
<td>CELSIZ</td>
<td>Length and width of each cell in the grid plane</td>
<td>Inches</td>
</tr>
<tr>
<td>D1</td>
<td></td>
<td>Distance from the first intersect on the target to the center plane of the target</td>
<td>Inches</td>
</tr>
<tr>
<td>D2</td>
<td></td>
<td>Distance from the last intersect on the target to the center plane of the target</td>
<td>Inches</td>
</tr>
<tr>
<td>DIST</td>
<td></td>
<td>Distance from the start of a new region through the region until a new region is encountered</td>
<td>Inches</td>
</tr>
<tr>
<td>H</td>
<td>H</td>
<td>Horizontal distance from center of grid plane to random point in specified grid cell</td>
<td>Inches</td>
</tr>
<tr>
<td>HREF</td>
<td>H ref</td>
<td>Horizontal distance from the center of the grid plane to the center of the grid square</td>
<td>Inches</td>
</tr>
<tr>
<td>I15</td>
<td></td>
<td>Value 215. Used for packing and unpacking data in a single word</td>
<td>ND</td>
</tr>
<tr>
<td>I30</td>
<td></td>
<td>Value 230. Used for packing and unpacking data in a single word</td>
<td>ND</td>
</tr>
<tr>
<td>IA(9)</td>
<td></td>
<td>Array for entering the logical operator or when entering region data during Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IC(4)</td>
<td>---</td>
<td>Array for entering alphanemic program control data and the abbreviation of the body type during Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>IENC</td>
<td>---</td>
<td>Region number enclosing the target and attack plane</td>
<td>ND</td>
</tr>
<tr>
<td>IENTLV</td>
<td>---</td>
<td>Option variable used to determine if Subroutine GENI is to print out the region enter/leave tables</td>
<td>ND</td>
</tr>
<tr>
<td>IERR</td>
<td>---</td>
<td>Variable used to count the number of errors in the geometry input. Also used to count the number of errors in Subroutine Gl</td>
<td>ND</td>
</tr>
<tr>
<td>IERRO</td>
<td>---</td>
<td>Counter for the number of O component code errors</td>
<td>ND</td>
</tr>
<tr>
<td>IGRID</td>
<td>---</td>
<td>Grid square of the origin of the current ray</td>
<td>ND</td>
</tr>
<tr>
<td>IN(9)</td>
<td>---</td>
<td>Array for entering the operator (+ or -) and the body number when entering region data during Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>INORM</td>
<td>---</td>
<td>Control variable for Subroutine ARS to either compute the normal distance (INORM=1), or to compute the line-of-sight distance (INORM=0)</td>
<td>ND</td>
</tr>
<tr>
<td>IR</td>
<td>---</td>
<td>Region number where the point along the ray is presently located</td>
<td>ND</td>
</tr>
<tr>
<td>IRANDM</td>
<td>---</td>
<td>Control variable passed from MAIN to Function RAN for computing a random number</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IRAYSK</td>
<td>---</td>
<td>Control variable passed from MAIN to Subroutine GRID to skip a random number of cells in the grid plane if IRAYSK is not equal to zero</td>
<td>ND</td>
</tr>
<tr>
<td>IRN</td>
<td>---</td>
<td>Region number of region type data entered</td>
<td>ND</td>
</tr>
<tr>
<td>IRSTRT</td>
<td>---</td>
<td>Region number of ray origin</td>
<td>ND</td>
</tr>
<tr>
<td>IT(10)</td>
<td>---</td>
<td>Title of the problem of up to 60 alphanumeric characters</td>
<td>ND</td>
</tr>
<tr>
<td>ITAPE8</td>
<td>---</td>
<td>Option variable for suppressing printout (ITAPE8=0)</td>
<td>ND</td>
</tr>
<tr>
<td>ITESTG</td>
<td>---</td>
<td>Option variable used to determine if Subroutine TESTG is to be called by the MAIN program</td>
<td>ND</td>
</tr>
<tr>
<td>ITR(200)</td>
<td>---</td>
<td>Storage array for recording ray contact data: surface number, body number, next region number</td>
<td>ND</td>
</tr>
<tr>
<td>IVIH</td>
<td>---</td>
<td>Two-digit random number computed in Subroutine GRID for printout by Subroutine TRACK</td>
<td>ND</td>
</tr>
<tr>
<td>IVOLUM</td>
<td>---</td>
<td>Option variable used to determine if Subroutine VOLUM is to be called by the MAIN program</td>
<td>ND</td>
</tr>
<tr>
<td>IWOT</td>
<td>---</td>
<td>Option variable used to determine if Subroutine GRID data is to be written on output tape 1</td>
<td>ND</td>
</tr>
</tbody>
</table>
### List of Symbols and Abbreviations

#### (Simulation Model)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IYES</td>
<td>---</td>
<td>Integer variable equal to one which is used for option testing</td>
<td>ND</td>
</tr>
<tr>
<td>KLOOP</td>
<td>---</td>
<td>Internal ray counter for keeping track of each new ray in Subroutine Gl</td>
<td>ND</td>
</tr>
<tr>
<td>KHIT</td>
<td>---</td>
<td>Counter for keeping track of the number of components hit along a given ray</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Counter for the number of intersects along a given ray</td>
<td>ND</td>
</tr>
<tr>
<td>LABUT</td>
<td>---</td>
<td>Location of the beginning abutting RPP data in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LBASE</td>
<td>---</td>
<td>Beginning location of the MASTER-ASTER array (usually one)</td>
<td>ND</td>
</tr>
<tr>
<td>LBODY</td>
<td>---</td>
<td>Location of the body pointers in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LDATA</td>
<td>---</td>
<td>Temporary address of data in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LEGEOM</td>
<td>---</td>
<td>Location of the end of the geometry data processed by Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>LIO</td>
<td>---</td>
<td>Location of the beginning of a temporary storage area in the MASTER-ASTER array for use by Subroutine Gl</td>
<td>ND</td>
</tr>
<tr>
<td>LIRFO</td>
<td>---</td>
<td>Starting location of the region ID data in the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LMAX</td>
<td>---</td>
<td>Total number of intersects that occur along a given ray</td>
<td>ND</td>
</tr>
<tr>
<td>LOCDA</td>
<td>---</td>
<td>Location of data in the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>LOOP</td>
<td>---</td>
<td>Value of KLOOP when ray data was stored</td>
<td>ND</td>
</tr>
<tr>
<td>LREGD</td>
<td>---</td>
<td>Beginning location in the MASTER-ASTER array of the region data pointer/number of bodies in region</td>
<td>ND</td>
</tr>
<tr>
<td>LRI</td>
<td>---</td>
<td>Surface number of the entering intersect</td>
<td>ND</td>
</tr>
<tr>
<td>LRIN</td>
<td>---</td>
<td>Beginning location in the ASTER array of the RIN data</td>
<td>ND</td>
</tr>
<tr>
<td>LRO</td>
<td>---</td>
<td>Surface number of the exit intersect</td>
<td>ND</td>
</tr>
<tr>
<td>LROT</td>
<td>---</td>
<td>Beginning location in the ASTER array of the ROUT data</td>
<td>ND</td>
</tr>
<tr>
<td>LRPPD</td>
<td>---</td>
<td>Beginning location in the ASTER array of the RPP minimum/maximum values</td>
<td>ND</td>
</tr>
<tr>
<td>LSCAL</td>
<td>---</td>
<td>Beginning location in the ASTER array where the scalar data is to be entered by Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>LSURF</td>
<td>---</td>
<td>Surface number of body surface hit (negative if exit intersect)</td>
<td>ND</td>
</tr>
<tr>
<td>LTRIP</td>
<td>---</td>
<td>Beginning location in the ASTER array where the triplet data is to be entered by Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>NASC</td>
<td>---</td>
<td>Current body number (-1 means start new ray, -2 means find normal distance)</td>
<td>ND</td>
</tr>
<tr>
<td>NBODY</td>
<td>---</td>
<td>Number of bodies used to describe the target geometry other than RPP's</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>NDQ</td>
<td>---</td>
<td>Upper limit or size in words of the MASTER-ASTER array, usually 10,000</td>
<td>ND</td>
</tr>
<tr>
<td>NG1ERR</td>
<td>---</td>
<td>Maximum number of errors that are allowed in Subroutine G1</td>
<td>ND</td>
</tr>
<tr>
<td>NIR</td>
<td>---</td>
<td>Region identification (region component code)</td>
<td>ND</td>
</tr>
<tr>
<td>NN</td>
<td>---</td>
<td>Variable made up of the body number plus the number of RPP's</td>
<td>ND</td>
</tr>
<tr>
<td>NO</td>
<td>---</td>
<td>Integer variable equal to zero which is used for option testing</td>
<td>ND</td>
</tr>
<tr>
<td>NRMAX</td>
<td>---</td>
<td>Total number of regions used to describe the target geometry</td>
<td>ND</td>
</tr>
<tr>
<td>NRPP</td>
<td>---</td>
<td>Number of rectangular parallelepipeds used to enclose the target geometry</td>
<td>ND</td>
</tr>
<tr>
<td>NTYPE</td>
<td>---</td>
<td>Space code of the region following the next intersect</td>
<td>ND</td>
</tr>
<tr>
<td>PINF</td>
<td>---</td>
<td>Value $10^{50}$. Used to represent infinity</td>
<td>ND</td>
</tr>
<tr>
<td>RIN</td>
<td>RIN</td>
<td>Distance along the ray from the beginning of a given region to the entry intersect of the body under consideration</td>
<td>Inches</td>
</tr>
<tr>
<td>ROUT</td>
<td>ROUT</td>
<td>Distance along the ray from the beginning of a given region to the exit intersect of the body under consideration</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>SA</td>
<td>$\sin \alpha$</td>
<td>Sine of the azimuth angle of the ray with respect to the target geometry</td>
<td>ND</td>
</tr>
<tr>
<td>SE</td>
<td>$\sin \theta$</td>
<td>Sine of the elevation angle of the ray with respect to the target geometry</td>
<td>ND</td>
</tr>
<tr>
<td>SLOS</td>
<td>---</td>
<td>Line-of-sight distance through region following present intersect</td>
<td>Inches</td>
</tr>
<tr>
<td>SN</td>
<td>---</td>
<td>Normal distance through a given region</td>
<td>Inches</td>
</tr>
<tr>
<td>SSPC</td>
<td>---</td>
<td>Line-of-sight distance through space following a given intersect</td>
<td>Inches</td>
</tr>
<tr>
<td>TR(200)</td>
<td>---</td>
<td>Storage array for recording ray line-of-sight distance from contact to contact</td>
<td>ND</td>
</tr>
<tr>
<td>TRAVEL</td>
<td>---</td>
<td>Line-of-sight distance from origin of ray to present intersect</td>
<td>Inches</td>
</tr>
<tr>
<td>V</td>
<td>$V$</td>
<td>Vertical distance from center of grid plane to random point in specified grid cell</td>
<td>Inches</td>
</tr>
<tr>
<td>VREF</td>
<td>$V_{ref}$</td>
<td>Vertical distance from center of grid plane to center of grid square</td>
<td>Inches</td>
</tr>
<tr>
<td>WB(3)</td>
<td>$\overline{WB}$</td>
<td>Present direction cosines of the ray</td>
<td>ND</td>
</tr>
<tr>
<td>WS(3)</td>
<td>$\overline{WS}$</td>
<td>Original direction cosines of the ray</td>
<td>ND</td>
</tr>
<tr>
<td>X(6)</td>
<td>---</td>
<td>Temporary storage array for entering the six bounding planes of an RPP</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>$\text{XB}(3)$</td>
<td>$\text{XB}$</td>
<td>$x, y, z$ coordinates of the origin of the ray with respect to a given intersect</td>
<td>Inches</td>
</tr>
<tr>
<td>$\text{XBS}(3)$</td>
<td>$\bar{x}_p$</td>
<td>$x, y, z$ coordinates of a point in a plane through the center of the target geometry through which the ray will pass</td>
<td>Inches</td>
</tr>
<tr>
<td>$\text{XS}(3)$</td>
<td>$---$</td>
<td>$x, y, z$ coordinates of the origin of the ray in the grid plane</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>---</td>
<td>Six-element array for entering a description of the region type data</td>
<td>ND</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index in DO loops or for entering and writing out data</td>
<td>ND</td>
</tr>
<tr>
<td>ICODE</td>
<td>---</td>
<td>Item code of component</td>
<td>ND</td>
</tr>
<tr>
<td>IDENT</td>
<td>---</td>
<td>Space code and special identification of region</td>
<td>ND</td>
</tr>
<tr>
<td>IRDTP4</td>
<td>---</td>
<td>Option variable for entering the target geometry data from tape</td>
<td>ND</td>
</tr>
<tr>
<td>IRN</td>
<td>---</td>
<td>Area for entering region numbers when entering region data</td>
<td>ND</td>
</tr>
<tr>
<td>IWRTP4</td>
<td>---</td>
<td>Option variable for writing out the target geometry data onto tape</td>
<td>ND</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>Location pointer to the MASTER array for storing ICODE and IDENT region data</td>
<td>ND</td>
</tr>
<tr>
<td>NAREA</td>
<td>---</td>
<td>Number of aspect angles to be processed by Subroutine AREA</td>
<td>ND</td>
</tr>
<tr>
<td>NOAA</td>
<td>---</td>
<td>Number of aspect angles to be processed by Subroutine GRID</td>
<td>ND</td>
</tr>
</tbody>
</table>
**LIST OF SYMBOLS AND ABBREVIATIONS**  
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>Distance from the center to one end of an ellipsoid of revolution</td>
<td>Inches</td>
</tr>
<tr>
<td>ASQ</td>
<td></td>
<td>Distance squared from center to one end of an ellipsoid of revolution</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td>Distance from the center to a focus of an ellipsoid of revolution</td>
<td>Inches</td>
</tr>
<tr>
<td>CX</td>
<td></td>
<td>x component distance from the center to a focus of an ellipsoid of revolution</td>
<td>Inches</td>
</tr>
<tr>
<td>CY</td>
<td></td>
<td>y component distance from the center to a focus of an ellipsoid of revolution</td>
<td>Inches</td>
</tr>
<tr>
<td>CZ</td>
<td></td>
<td>z component distance from the center of a focus of an ellipsoid of revolution</td>
<td>Inches</td>
</tr>
<tr>
<td>FX(20)</td>
<td></td>
<td>Temporary array used to enter body triplet and scalar data. Also used for manipulating and computing additional data before storing the data into the MASTER-ASTER array in its final format</td>
<td>ND</td>
</tr>
<tr>
<td>HDN</td>
<td></td>
<td>Dot product of the height vector and the normal to the base ellipse of a truncated elliptic cone</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>Index for various DO loops</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td></td>
<td>Pointer to the first position of a group of data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>I2</td>
<td></td>
<td>Pointer to the last position of a group of data in the ASTER array</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
#### (SIMULATION MODEL)

#### SUBROUTINE GENI (Continued)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAA(1)</td>
<td></td>
<td>Storage location containing the Hollerith logical operator &quot;bbb&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAA(2)</td>
<td></td>
<td>Storage location containing the Hollerith logical operator &quot;OBb&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAA(3)</td>
<td></td>
<td>Storage location containing the Hollerith logical operator &quot;bRb&quot;</td>
<td>ND</td>
</tr>
</tbody>
</table>

(Table continued on next page)
<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IAA(4)</td>
<td>---</td>
<td>Storage location containing the Hollerith logical operator &quot;Rbb&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAA(5)</td>
<td>---</td>
<td>Storage location containing the Hollerith logical operator &quot;RAB&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAA(6)</td>
<td>---</td>
<td>Storage location containing the Hollerith logical operator &quot;ARb&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAA(7)</td>
<td>---</td>
<td>Storage location containing the Hollerith logical operator &quot;bAb&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAA(8)</td>
<td>---</td>
<td>Storage location containing the Hollerith logical operator &quot;Abb&quot;</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(1)</td>
<td>---</td>
<td>Storage location containing the integer 4 for converting the Hollerith logical operator to a numerical value of four</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(2)</td>
<td>---</td>
<td>Storage location containing the integer 1 for converting the Hollerith logical operator to a numerical value of one</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(3)</td>
<td>---</td>
<td>Storage location containing the integer 1 for converting the Hollerith logical operator to a numerical value of one</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(4)</td>
<td>---</td>
<td>Storage location containing the integer 1 for converting the Hollerith logical operator to a numerical value of one</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(5)</td>
<td>---</td>
<td>Storage location containing the integer 2 for converting the Hollerith logical operator to a numerical value of two</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(6)</td>
<td>---</td>
<td>Storage location containing the integer 2 for converting the Hollerith logical operator to a numerical value of two</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>IAN(7)</td>
<td>---</td>
<td>Storage location containing the integer 3 for converting the Hollerith logical operator to a numerical value of three</td>
<td>ND</td>
</tr>
<tr>
<td>IAN(8)</td>
<td>---</td>
<td>Storage location containing the integer 3 for converting the Hollerith logical operator to a numerical value of three</td>
<td>ND</td>
</tr>
<tr>
<td>IBL</td>
<td>---</td>
<td>Storage location containing one Hollerith blank for testing for blank fields on card input</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>---</td>
<td>Number of bodies in the larger of two regions that is used as the upper limit of a DO loop for comparing each item in the smaller region with each item in the larger region</td>
<td>ND</td>
</tr>
<tr>
<td>IK</td>
<td>---</td>
<td>Lower limit of a DO loop used to print out the MASTER-ASTER array three words at a time where IK is the location of the first of the three words</td>
<td>ND</td>
</tr>
<tr>
<td>IK2</td>
<td>---</td>
<td>Upper limit of a DO loop used to print out the MASTER-ASTER array three words at a time where IK2 is the location of the third of three words</td>
<td>ND</td>
</tr>
<tr>
<td>IO</td>
<td>---</td>
<td>Number of bodies in the smaller of two regions that is used as the upper limit of a DO loop for comparing each item in the smaller region with each item in the larger region</td>
<td>ND</td>
</tr>
<tr>
<td>IOP</td>
<td>---</td>
<td>Logical operator of the body to be tested or compared from the enter/leave tables in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IOPER</td>
<td>---</td>
<td>Refer to IOP</td>
<td>ND</td>
</tr>
<tr>
<td>IOPI</td>
<td>---</td>
<td>Refer to IOP</td>
<td>ND</td>
</tr>
<tr>
<td>IOPO</td>
<td>---</td>
<td>Refer to IOP</td>
<td>ND</td>
</tr>
<tr>
<td>IPRIN</td>
<td>---</td>
<td>Option control variable for printing out the entire MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IR</td>
<td>---</td>
<td>Number of the region entered from the card input</td>
<td>ND</td>
</tr>
<tr>
<td>IRCHEK</td>
<td>---</td>
<td>Option control variable for verifying the validity of the region enter/leave tables</td>
<td>ND</td>
</tr>
<tr>
<td>IS</td>
<td>---</td>
<td>Value of +1 or -1 used in preparing the enter/leave tables. For IS = -1 the leave table is prepared and for IS = +1 the enter table is prepared</td>
<td>ND</td>
</tr>
<tr>
<td>ITEMP</td>
<td>---</td>
<td>Variable used to represent the location of a region data pointer word</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(1)</td>
<td>---</td>
<td>Storage location containing the Hollerith string BOX to represent a box</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(2)</td>
<td>---</td>
<td>Storage location containing the Hollerith string SPH to represent a sphere</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(3)</td>
<td>---</td>
<td>Storage location containing the Hollerith string RCC to represent a right circular cylinder</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(4)</td>
<td>---</td>
<td>Storage location containing the Hollerith string REC to represent a right elliptic cylinder</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS  
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITY(5)</td>
<td>---</td>
<td>Storage location containing the Hollerith string TRC to represent a right truncated cone</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(6)</td>
<td>---</td>
<td>Storage location containing the Hollerith string ELL to represent an ellipsoid of revolution</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(7)</td>
<td>---</td>
<td>Storage location containing the Hollerith string RAW to represent a right angle wedge</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(8)</td>
<td>---</td>
<td>Storage location containing the Hollerith string ARB to represent an arbitrary polyhedron</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(9)</td>
<td>---</td>
<td>Storage location containing the Hollerith string TEC to represent a truncated elliptic cone</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(10)</td>
<td>---</td>
<td>Storage location containing the Hollerith string TOR to represent a torus</td>
<td>ND</td>
</tr>
<tr>
<td>ITY(11)</td>
<td>---</td>
<td>Storage location containing the Hollerith string ARS to represent an arbitrary surface</td>
<td>ND</td>
</tr>
<tr>
<td>ITYPE</td>
<td>---</td>
<td>Variable first used to store the string of the body type when a card is entered. It is later used to store the integer equivalent of the body type</td>
<td>ND</td>
</tr>
<tr>
<td>IWH</td>
<td>---</td>
<td>Pointer to the location of pointer data in the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Index used to represent region numbers</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>J1-J12</td>
<td>---</td>
<td>Variables used to represent the first and last location of either the region enter table or region leave table during the printout of these tables</td>
<td>ND</td>
</tr>
<tr>
<td>JJ</td>
<td>---</td>
<td>Variable used as an index when comparing the region data of two different regions</td>
<td>ND</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>Index for use in DO loops and in entering data</td>
<td>ND</td>
</tr>
<tr>
<td>KI</td>
<td>---</td>
<td>Index for the number of bodies in the larger of two regions where the region tables are being verified</td>
<td>ND</td>
</tr>
<tr>
<td>KLK</td>
<td>---</td>
<td>Variable used to represent the location of a region data word</td>
<td>ND</td>
</tr>
<tr>
<td>KO</td>
<td>---</td>
<td>Index for the number of bodies in the smaller of two regions where the region tables are being verified</td>
<td>ND</td>
</tr>
<tr>
<td>KRI</td>
<td>---</td>
<td>Variable used to represent the location of a region data pointer word</td>
<td>ND</td>
</tr>
<tr>
<td>KRJ</td>
<td>---</td>
<td>Variable used to represent the location of a region data pointer word</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Temporary storage location or coordinate index</td>
<td>ND</td>
</tr>
<tr>
<td>L1</td>
<td>---</td>
<td>Variable used to represent the location of the last word in the enter/leave table</td>
<td>ND</td>
</tr>
<tr>
<td>LAR</td>
<td>---</td>
<td>Variable used to represent the last location of RPP data in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>LBOD</td>
<td>---</td>
<td>Variable used to represent the beginning location of the body data pointers</td>
<td>ND</td>
</tr>
<tr>
<td>LBOT</td>
<td>---</td>
<td>Variable initially used to represent the beginning location of the scalar data. This value changes as additional body data is stored backward from the beginning location of the scalar data</td>
<td>ND</td>
</tr>
<tr>
<td>LD</td>
<td>---</td>
<td>Variable used to represent the location of the last word of the body data pointers</td>
<td>ND</td>
</tr>
<tr>
<td>LE</td>
<td>---</td>
<td>Variable used to represent the number of data elements or pointers for a given body</td>
<td>ND</td>
</tr>
<tr>
<td>LEAV</td>
<td>---</td>
<td>Pointer to the first location of the region leave table</td>
<td>ND</td>
</tr>
<tr>
<td>LEGEOM</td>
<td>---</td>
<td>Pointer to the last location of the geometry data processed by Subroutine GENI</td>
<td>ND</td>
</tr>
<tr>
<td>LENLV</td>
<td>---</td>
<td>Variable used to represent the beginning location of the leave/enter tables</td>
<td>ND</td>
</tr>
<tr>
<td>LENT</td>
<td>---</td>
<td>Pointer to the first location of the region enter table</td>
<td>ND</td>
</tr>
<tr>
<td>LL</td>
<td>---</td>
<td>Variable used as a counter to represent the number of errors in the region data when the checking option is performed</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Variable used to represent the location of a specific word in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOCI</td>
<td>---</td>
<td>Variable used to represent the location of the region data for a specific region</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>LOCJ</td>
<td>---</td>
<td>Variable used to represent the location of the region data for a specific region</td>
<td>ND</td>
</tr>
<tr>
<td>LREGL</td>
<td>---</td>
<td>Variable used to represent the beginning location of the region data</td>
<td>ND</td>
</tr>
<tr>
<td>LSI</td>
<td>---</td>
<td>Variable used to indicate whether triplet data (LSI=0) or scalar data (LSI=1) is to be stored by Subroutine SEE3</td>
<td>ND</td>
</tr>
<tr>
<td>LSUB</td>
<td>---</td>
<td>Variable used to represent the number of unused words between the last pointer word and the first body data word</td>
<td>ND</td>
</tr>
<tr>
<td>LT</td>
<td>---</td>
<td>Pointer to a reference location for storing pointers to triplet data</td>
<td>ND</td>
</tr>
<tr>
<td>M</td>
<td>---</td>
<td>Counter or pointer for the region data in the body pointer section of the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>MIS</td>
<td>---</td>
<td>Counter for the number of matching operator/body combinations in another region</td>
<td>ND</td>
</tr>
<tr>
<td>MM</td>
<td>---</td>
<td>Variable used to index to specific body data words in the region data</td>
<td>ND</td>
</tr>
<tr>
<td>MMM</td>
<td>---</td>
<td>Index used to prepare either the region leaving table or region entering table</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Index used to represent body number or region number</td>
<td>ND</td>
</tr>
<tr>
<td>NBI</td>
<td>---</td>
<td>Region table body number to be compared</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>NBNR</td>
<td>---</td>
<td>Variable that represents the total number of geometric shapes used to describe the target geometry. It is used in printing the enter/leave table.</td>
<td>ND</td>
</tr>
<tr>
<td>NBO</td>
<td>---</td>
<td>Region table body number under test</td>
<td>ND</td>
</tr>
<tr>
<td>NBOD(I)</td>
<td>---</td>
<td>Eleven-element array used to count the number of times each of the eleven body shapes was used to describe the target geometry.</td>
<td>ND</td>
</tr>
<tr>
<td>NC</td>
<td>---</td>
<td>Number of bodies in a given region</td>
<td>ND</td>
</tr>
<tr>
<td>NEAV</td>
<td>---</td>
<td>Number of bodies in a given region leave table</td>
<td>ND</td>
</tr>
<tr>
<td>NENT</td>
<td>---</td>
<td>Number of bodies in a given region enter table</td>
<td>ND</td>
</tr>
<tr>
<td>NO1(M)</td>
<td>---</td>
<td>Two three-element arrays used to represent the last 15 bits and the 15 bits previous to the last 15 bits respectively of a computer word during the MASTER-ASTER array printout option when three words are printed out at a time.</td>
<td>ND</td>
</tr>
<tr>
<td>NO2(M)</td>
<td>---</td>
<td>Three-element array used to represent the three locations of the words for printout during the MASTER-ASTER array printout option.</td>
<td>ND</td>
</tr>
<tr>
<td>NOO(M)</td>
<td>---</td>
<td></td>
<td>ND</td>
</tr>
<tr>
<td>NUM</td>
<td>---</td>
<td>Body number candidate for a region enter or leave table</td>
<td>ND</td>
</tr>
<tr>
<td>NUMI</td>
<td>---</td>
<td>Number of bodies in the smaller region when checking validity region data.</td>
<td>ND</td>
</tr>
<tr>
<td>NUMJ</td>
<td>---</td>
<td>Number of bodies in the larger region when checking validity of region data.</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>04(M)</td>
<td>---</td>
<td>Three element array used to store the three words to be printed out during the MASTER-ASTER array printout option</td>
<td>ND</td>
</tr>
<tr>
<td>TT(3)</td>
<td>---</td>
<td>Three element array used to store the x, y, and z coordinates of the normal to the base ellipse of a TEC</td>
<td>Inches</td>
</tr>
<tr>
<td>TT1(3)</td>
<td>---</td>
<td>Three element array used to store the x, y, and z coordinates of the semi-major axis of the base ellipse of a TEC</td>
<td>Inches</td>
</tr>
<tr>
<td>TT2(3)</td>
<td>---</td>
<td>Three element array used to store the x, y, and z coordinates of the semi-minor axis of the base ellipse of a TEC</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS

(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>---</td>
<td>Location of the pointers for a given side of an RPP, or Number of an RPP for use in DO loops</td>
<td>ND</td>
</tr>
<tr>
<td>I2</td>
<td>---</td>
<td>Number of abutting RPP's in packed format</td>
<td>ND</td>
</tr>
<tr>
<td>I5</td>
<td>---</td>
<td>Pointer to the boundary coordinate for a given side of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>I3</td>
<td>---</td>
<td>Pointer to the location of the boundary coordinate for a given side of an RPP, or Number of an RPP for use as the lower limit in a DO loop</td>
<td>ND</td>
</tr>
<tr>
<td>I6</td>
<td>---</td>
<td>Index used to represent the side or number of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>---</td>
<td>Location of the pointers for a given side of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Location of the pointers for a given side of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>JJ</td>
<td>---</td>
<td>Location of the pointers for a given side of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>Location of the pointers for a given side of an RPP, or Index used to represent side pairs of an RPP for use in a DO loop</td>
<td>ND</td>
</tr>
<tr>
<td>K2</td>
<td>---</td>
<td>Even numbered side of a given RPP</td>
<td>ND</td>
</tr>
<tr>
<td>K21</td>
<td>---</td>
<td>Odd numbered side of a given RPP</td>
<td>ND</td>
</tr>
<tr>
<td>K41</td>
<td>---</td>
<td>Number representing the sum of the numbers of two opposite sides of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Index for referencing storage in the LRPPD section</td>
<td>ND</td>
</tr>
<tr>
<td>LAR</td>
<td>---</td>
<td>Location of the last word of RPP data</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>LAST</td>
<td>---</td>
<td>Location of the last word of RPP data</td>
<td>ND</td>
</tr>
<tr>
<td>LL</td>
<td>---</td>
<td>Counter for the number of abutting RPP's for a given side of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>M</td>
<td>---</td>
<td>Switch used in determining in which part of a packed word the number of an abutting RPP is to be stored</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Represents the number of an RPP or the number of a side of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>NC</td>
<td>---</td>
<td>Represents the side of an RPP opposite the side under test</td>
<td>ND</td>
</tr>
<tr>
<td>NN</td>
<td>---</td>
<td>Sum of the numbers representing two opposite sides of an RPP</td>
<td>ND</td>
</tr>
<tr>
<td>NRPPL</td>
<td>---</td>
<td>One less than the number of RPP's used to represent the target geometry</td>
<td>ND</td>
</tr>
<tr>
<td>X(J)</td>
<td>---</td>
<td>Value of a boundary coordinate for a side of a given RPP</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## List of Symbols and Abbreviations

### Subroutine Albert

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>Coefficient of $x$ in the equation of the plane</td>
<td>ND</td>
</tr>
<tr>
<td>A2B2C2</td>
<td></td>
<td>Sum of the squares of the $x$, $y$, and $z$ components of the equation of the plane</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>AA(I,J)</td>
<td></td>
<td>24-element two-dimensional array for entering and storing the coordinates of the eight vertices of the ARB</td>
<td>Inches</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>Coefficient of $y$ in the equation of the plane</td>
<td>ND</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>Coefficient of $z$ in the equation of the plane</td>
<td>ND</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
<td>Constant term in the equation of the plane</td>
<td>ND</td>
</tr>
<tr>
<td>D12</td>
<td></td>
<td>Square of the length between two vertices of a side of the ARB</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>D1210</td>
<td></td>
<td>$D12 \times 10^{-12}$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>D2</td>
<td></td>
<td>Perpendicular distance from the fourth vertex to the plane formed by the first three vertices</td>
<td>Inches</td>
</tr>
<tr>
<td>D22</td>
<td></td>
<td>$(D2)^2$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>F(4)</td>
<td></td>
<td>Four-element array for storing the results of the four vertices not part of the plane under consideration when substituted into the plane equation</td>
<td>ND</td>
</tr>
<tr>
<td>FX</td>
<td></td>
<td>Six-element array that contains the coordinates of the first two vertices of the ARB</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>Index for entering the coordinates of the vertices of the ARB</td>
<td>ND</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS

### (SIMULATION MODEL)

#### SUBROUTINE ALBERT (Continued)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IA</td>
<td>---</td>
<td>24-element two-dimensional array for entering and storing the four ordinal vertex numbers for six planes of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>IC</td>
<td>---</td>
<td>Fourth ordinal number of one of the vertices of a plane</td>
<td>ND</td>
</tr>
<tr>
<td>IWH</td>
<td>---</td>
<td>Pointer returned by Subroutine SEE3 to the location where the ARB data was stored</td>
<td>ND</td>
</tr>
<tr>
<td>IX</td>
<td>---</td>
<td>First ordinal number of one of the vertices of a plane</td>
<td>ND</td>
</tr>
<tr>
<td>IY</td>
<td>---</td>
<td>Second ordinal number of one of the vertices of a plane</td>
<td>ND</td>
</tr>
<tr>
<td>IZ</td>
<td>---</td>
<td>Third ordinal number of one of the vertices of a plane</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Index in a DO loop for storing coordinates of two vertices. Also used to count the number of vertices on the positive side of the plane under test</td>
<td>ND</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>Index for referencing elements in array FX. Also used as an index to zero array F</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Index for storing data into the F array</td>
<td>ND</td>
</tr>
<tr>
<td>LBOT</td>
<td>---</td>
<td>Pointer to the location where data was last stored by Subroutine SEE3</td>
<td>ND</td>
</tr>
<tr>
<td>LS1</td>
<td>---</td>
<td>Variable used to indicate to Subroutine SEE3 whether triplet data (LS1=0) or scalar data (LS1=1) is to be stored</td>
<td>ND</td>
</tr>
<tr>
<td>M</td>
<td>---</td>
<td>Variable for counting the number of vertices on the negative side of the plane under test</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Variable for counting the number of vertices remaining on the plane under test</td>
<td>ND</td>
</tr>
<tr>
<td>NDQ</td>
<td>---</td>
<td>Upper limit pointer of the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>S</td>
<td>---</td>
<td>Square root of the sum of the squares of the x, y, and z coefficients</td>
<td>Inches</td>
</tr>
<tr>
<td>WX</td>
<td>---</td>
<td>x direction cosine of a vector from the origin perpendicular to the plane</td>
<td>Inches</td>
</tr>
<tr>
<td>WY</td>
<td>---</td>
<td>y direction cosine of a vector from the origin perpendicular to the plane</td>
<td>Inches</td>
</tr>
<tr>
<td>WZ</td>
<td>---</td>
<td>z direction cosine of a vector from the origin perpendicular to the plane</td>
<td>Inches</td>
</tr>
<tr>
<td>X1</td>
<td>---</td>
<td>x coordinate of the first vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>X2</td>
<td>---</td>
<td>x coordinate of the second vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>X3</td>
<td>---</td>
<td>x coordinate of the third vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>X4</td>
<td>---</td>
<td>x coordinate of the fourth vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Y1</td>
<td>---</td>
<td>y coordinate of the first vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Y2</td>
<td>---</td>
<td>y coordinate of the second vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Y3</td>
<td>---</td>
<td>y coordinate of the third vertex of a plane</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS (SIMULATION MODEL)

**SUBROUTINE ALBERT (Concluded)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y4</td>
<td>---</td>
<td>y coordinate of the fourth vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Z1</td>
<td>---</td>
<td>z coordinate of the first vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Z2</td>
<td>---</td>
<td>z coordinate of the second vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Z3</td>
<td>---</td>
<td>z coordinate of the third vertex of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Z4</td>
<td>---</td>
<td>z coordinate of the fourth vertex of a plane</td>
<td>Inches</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS

## (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>---</td>
<td>DO loop index equal to curve number during input of ARS point data</td>
<td>ND*</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>DO loop index for referencing storage locations in the ARS point data section</td>
<td>ND</td>
</tr>
<tr>
<td>LBOT</td>
<td>---</td>
<td>Pointer to beginning location of storage area for ARS</td>
<td>ND</td>
</tr>
<tr>
<td>LDATA</td>
<td>---</td>
<td>Pointer to next available location in body pointer section of MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Storage location pointer for ARS point data section</td>
<td>ND</td>
</tr>
<tr>
<td>LOCC</td>
<td>---</td>
<td>Storage location pointer for ARS point data section</td>
<td>ND</td>
</tr>
<tr>
<td>L1</td>
<td>---</td>
<td>DO loops lower limit for referencing storage locations in ARS point data section</td>
<td>ND</td>
</tr>
<tr>
<td>L2</td>
<td>---</td>
<td>DO loops upper limit for referencing storage locations in ARS point data section</td>
<td>ND</td>
</tr>
<tr>
<td>M</td>
<td>---</td>
<td>Number of curves used to describe given ARS</td>
<td>ND</td>
</tr>
<tr>
<td>MN</td>
<td>---</td>
<td>Total number of points used to describe given ARS</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Number of points per curve</td>
<td>ND</td>
</tr>
<tr>
<td>NP</td>
<td>---</td>
<td>Number of points to be stored; NP=2<em>N</em>(M-1); points stored in pairs between consecutive curves</td>
<td>ND</td>
</tr>
<tr>
<td>NSTR</td>
<td>---</td>
<td>Total number of storage words required for the given ARS</td>
<td>ND</td>
</tr>
<tr>
<td>NT</td>
<td>---</td>
<td>Number of triangles described</td>
<td>ND</td>
</tr>
</tbody>
</table>

*Non-dimensional*
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

#### SUBROUTINE ARIN (Concluded)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>NT</td>
<td>---</td>
<td>Number of non-degenerate triangles in given ARS</td>
</tr>
<tr>
<td>UW(3)</td>
<td>---</td>
<td>x, y, z coordinates of vector between first point and second point of given triangle of ARS</td>
</tr>
<tr>
<td>VW(3)</td>
<td>---</td>
<td>x, y, z coordinates of vector between first point and third point of given triangle of ARS</td>
</tr>
<tr>
<td>W(3)</td>
<td>---</td>
<td>x, y, z coordinates of first point of given triangle of ARS</td>
</tr>
<tr>
<td>WN(3)</td>
<td>---</td>
<td>x, y, z coordinates of vector formed from cross product of vectors UW(3) and VW(3)</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>FX</td>
<td>---</td>
<td>x coordinate of the triplet data, or the value of the scalar quantity</td>
<td>ND</td>
</tr>
<tr>
<td>FXX</td>
<td>---</td>
<td>y coordinate of the triplet data, or the value of the scalar quantity</td>
<td>ND</td>
</tr>
<tr>
<td>FXXX</td>
<td>---</td>
<td>z coordinate of the triplet data, or the value of the scalar quantity</td>
<td>ND</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index for searching through the triplet/scalar data</td>
<td>ND</td>
</tr>
<tr>
<td>IWH</td>
<td>---</td>
<td>Pointer to location of the triplet or scalar data</td>
<td>ND</td>
</tr>
<tr>
<td>LBOT</td>
<td>---</td>
<td>Beginning locations of the triplet/scalar data section</td>
<td>ND</td>
</tr>
<tr>
<td>LDATA</td>
<td>---</td>
<td>Pointer to the next available location in the body section of the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LS1</td>
<td>---</td>
<td>Variable used to indicate whether triplet data (LS1=0) or scalar data (LS1=1) is in the argument list</td>
<td>ND</td>
</tr>
<tr>
<td>NDQ</td>
<td>---</td>
<td>Last location of the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>NDQ2</td>
<td>---</td>
<td>NDQ-2</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>Azimuth angle</td>
<td>Degrees</td>
</tr>
<tr>
<td>AR</td>
<td>α</td>
<td>Azimuth angle</td>
<td>Radians</td>
</tr>
<tr>
<td>CELL2</td>
<td>D/2</td>
<td>One-half the total number of grid cells</td>
<td>ND</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>Elevation angle</td>
<td>Degrees</td>
</tr>
<tr>
<td>ENGTH</td>
<td>ENGTH</td>
<td>Back-off distance of the shifted point of the given cell</td>
<td>Inches</td>
</tr>
<tr>
<td>ER</td>
<td>θ</td>
<td>Elevation angle</td>
<td>Radians</td>
</tr>
<tr>
<td>GROUND</td>
<td>---</td>
<td>z-coordinate of ground level</td>
<td>Inches</td>
</tr>
<tr>
<td>ICENTR</td>
<td>---</td>
<td>Control variable for originating the ray from the center of the cell when equal to one</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>Iₜ</td>
<td>Random number for computing a random horizontal point within a given cell</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>Iₐ</td>
<td>Variable that represents the row number of the grid</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>Iₐ</td>
<td>Random number for computing a random vertical point within a given cell</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>Jₐ</td>
<td>Variable that represents the column number of the grid</td>
<td>ND</td>
</tr>
<tr>
<td>KK</td>
<td>kₐ</td>
<td>Index of the major DO loop that represents the cell number</td>
<td>ND</td>
</tr>
<tr>
<td>KK1</td>
<td>---</td>
<td>Index that represents an x, y, or z coordinate in a DO loop</td>
<td>ND</td>
</tr>
<tr>
<td>MSHIFT</td>
<td>---</td>
<td>Random number between 0 and 24 that determines the random number of cells to be skipped</td>
<td>ND</td>
</tr>
<tr>
<td>NEND</td>
<td>Nₐ</td>
<td>Number of the last cell in the grid</td>
<td>ND</td>
</tr>
<tr>
<td>NSTART</td>
<td>---</td>
<td>Starting cell number, usually the first cell in the grid</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>NX</td>
<td>N (x)</td>
<td>Number of horizontal cells in the grid plane</td>
<td>ND</td>
</tr>
<tr>
<td>NY</td>
<td>N (y)</td>
<td>Number of vertical cells in the grid plane</td>
<td>ND</td>
</tr>
<tr>
<td>RADIAN</td>
<td>R</td>
<td>One degree in radians = 0.017453292519943</td>
<td>Radians</td>
</tr>
<tr>
<td>WP(3)</td>
<td>---</td>
<td>(x, y, z) coordinates of random direction cosines returned by Subroutine TROPIC</td>
<td>Inches</td>
</tr>
<tr>
<td>XSHIFT</td>
<td>XSHIFT</td>
<td>Distance target origin and grid plane center is effectively shifted in the X direction</td>
<td>Inches</td>
</tr>
<tr>
<td>YSHIFT</td>
<td>YSHIFT</td>
<td>Distance target origin and grid plane center is effectively shifted in the Y direction</td>
<td>Inches</td>
</tr>
<tr>
<td>ZSHIFT</td>
<td>ZSHIFT</td>
<td>Distance target origin and grid plane center is effectively shifted in the Z direction</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
**(SIMULATION MODEL)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANGLEl</td>
<td>---</td>
<td>Save area for obliquity angle of intersect for first half of output line</td>
<td>ND</td>
</tr>
<tr>
<td>ERROR</td>
<td>---</td>
<td>Two-element array containing Hollerith data for possible printout</td>
<td>ND</td>
</tr>
<tr>
<td>D1</td>
<td>---</td>
<td>Distance from the first intersect of the target to the center plane of the target</td>
<td>Inches</td>
</tr>
<tr>
<td>D2</td>
<td>---</td>
<td>Distance from the last intersect of the target to the center plane of the target</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index in a DO loop for referencing the TR and ITR arrays. Also used as an index for the x, y, and z coordinates in another DO loop</td>
<td>ND</td>
</tr>
<tr>
<td>I12</td>
<td>---</td>
<td>Value for packing data into array ITR. ( I12 = 2^{12} = 4096 )</td>
<td>ND</td>
</tr>
<tr>
<td>IDENT</td>
<td>---</td>
<td>Region identification space code</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>---</td>
<td>Horizontal cell number from center of grid</td>
<td>ND</td>
</tr>
<tr>
<td>IRPRIM</td>
<td>---</td>
<td>Region number returned by Subroutine G1</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Vertical cell number from center of grid</td>
<td>ND</td>
</tr>
<tr>
<td>JCNT</td>
<td>---</td>
<td>Counter used to count the spaces the ray encounters in the target</td>
<td>ND</td>
</tr>
<tr>
<td>JERRO</td>
<td>---</td>
<td>Index for ERROR array. Set to 2 if 0 component code error occurs.</td>
<td>ND</td>
</tr>
<tr>
<td>KLSURF</td>
<td>---</td>
<td>Surface number where the ray intersects the body (negative for exit intersect)</td>
<td>ND</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS

### SUBROUTINE TRACK (Continued) (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOC</td>
<td>---</td>
<td>Location of the region identity code in the region type data section</td>
<td>ND</td>
</tr>
<tr>
<td>MARMR</td>
<td>---</td>
<td>Flag for indicating armor material</td>
<td>ND</td>
</tr>
<tr>
<td>MSKRT</td>
<td>---</td>
<td>Flag for indicating skirt material</td>
<td>ND</td>
</tr>
<tr>
<td>MTARG</td>
<td>---</td>
<td>Flag for indicating target</td>
<td>ND</td>
</tr>
<tr>
<td>MVOL</td>
<td>---</td>
<td>Flag for indicating interior volume</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Number of components hit by ray</td>
<td>ND</td>
</tr>
<tr>
<td>NIR1</td>
<td>---</td>
<td>Save area for region identification (vehicle component) of intersect for first half of output line</td>
<td>ND</td>
</tr>
<tr>
<td>NTYPE1</td>
<td>---</td>
<td>Save area for the type of space following region of intersect for first half of output line</td>
<td>ND</td>
</tr>
<tr>
<td>S1</td>
<td>---</td>
<td>Distance to the next region returned by Subroutine G1</td>
<td>Inches</td>
</tr>
<tr>
<td>SLOSI1</td>
<td>---</td>
<td>Save area for the line-of-sight distance through region for first half of output line</td>
<td>ND</td>
</tr>
<tr>
<td>SN1</td>
<td>---</td>
<td>Save area for normal distance through region for first half of output line</td>
<td>ND</td>
</tr>
<tr>
<td>SPACE1</td>
<td>---</td>
<td>Save area for line-of-sight distance through space for first half of output line</td>
<td>ND</td>
</tr>
<tr>
<td>SUM</td>
<td>---</td>
<td>Summing location for computing distance from first target intersect to center plane of target</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>XP</td>
<td>---</td>
<td>x, y, and z coordinates of the new position of the ray returned by Subroutine Gl</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>Variable used in the ELL section to represent the length of the major axis</td>
<td>Inches</td>
</tr>
<tr>
<td>A1</td>
<td>a</td>
<td>Variable used in the REC section to represent the length from the center along the major axis</td>
<td>Inches</td>
</tr>
<tr>
<td>A2</td>
<td>b</td>
<td>Variable used in the REC section to represent the length from the center along the minor axis</td>
<td>Inches</td>
</tr>
<tr>
<td>ASQ</td>
<td>((R1/R2)^2) BSQ</td>
<td>Variable used in the TEC section to represent the square of the radius of the intersection ellipse along the semi-major axis</td>
<td>Inches $^2$</td>
</tr>
<tr>
<td>BSQ</td>
<td>([\gamma \cdot R4+R2(1-\gamma)]^2)</td>
<td>Variable used in the TEC section to represent the square of the radius of the intersection ellipse along the semi-minor axis</td>
<td>Inches $^2$</td>
</tr>
<tr>
<td>C</td>
<td>c</td>
<td>Variable used in the REC and TEC section to represent the distance from the center of the ellipse to the foci</td>
<td>Inches</td>
</tr>
<tr>
<td>DIS</td>
<td>---</td>
<td>Distance from ray origin to intersect in ARS section</td>
<td>Inches</td>
</tr>
<tr>
<td>DIV</td>
<td>(\sqrt{A^2+B^2+C^2})</td>
<td>Variable used in the ARB section to represent the square root of the sum of the squares of the x, y, and z coefficients of the equation of the intersected plane</td>
<td>Inches</td>
</tr>
<tr>
<td>GAMMA</td>
<td>(\frac{(X-V) \cdot N}{H \cdot N})</td>
<td>Variable used in the TEC section to represent the ratio of the height of the hit along the normal to the distance between the two planar surfaces</td>
<td>ND</td>
</tr>
</tbody>
</table>

545
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDN</td>
<td>$H \cdot N$</td>
<td>Variable used in the TEC section to represent the length of the height vector when projected onto the normal to the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>HF(3)</td>
<td>---</td>
<td>Array used in TEC section for the coordinates of the height vector</td>
<td>Inches</td>
</tr>
<tr>
<td>HH</td>
<td>$(X-V) \cdot N$</td>
<td>Variable used in the TEC section to represent the length of the vector from the vertex to the intersect when projected onto the normal to the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Variable used throughout the program as an index</td>
<td>ND</td>
</tr>
<tr>
<td>IDENT</td>
<td>---</td>
<td>Variable used to represent the space code of a particular region</td>
<td>ND</td>
</tr>
<tr>
<td>ITEMP</td>
<td>---</td>
<td>Temporary storage for entering the coordinate data for the box</td>
<td>ND</td>
</tr>
<tr>
<td>IJK</td>
<td>---</td>
<td>Variable used throughout the program as an index for retrieving data from the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IJK1</td>
<td>---</td>
<td>Refer to IJK</td>
<td>ND</td>
</tr>
<tr>
<td>IJK2</td>
<td>---</td>
<td>Refer to IJK</td>
<td>ND</td>
</tr>
<tr>
<td>IJK3</td>
<td>---</td>
<td>Refer to IJK</td>
<td>ND</td>
</tr>
<tr>
<td>ISPOT</td>
<td>---</td>
<td>Variable used as an index to locate specific region data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>ITYPE</td>
<td>---</td>
<td>Variable used to represent the body type of the intersected body</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Variable used throughout the program as an index</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>Variable used in the RAW section as an index to retrieve data from the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>KCOM</td>
<td>---</td>
<td>Variable used in the BOX section to test the surface number of the intersect for odd or even status</td>
<td>ND</td>
</tr>
<tr>
<td>L1</td>
<td>---</td>
<td>Pointer to the location of the x, y, and z coefficients for the equation of the intersected plane of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>LA</td>
<td>---</td>
<td>Pointer to the location of the direction cosines of the semi-major axis of the base ellipse of the TEC</td>
<td>ND</td>
</tr>
<tr>
<td>LH</td>
<td>---</td>
<td>Variable used in the BOX section as an index to locate the three ( H ) vectors of the BOX in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LK</td>
<td>---</td>
<td>Refer to K</td>
<td>ND</td>
</tr>
<tr>
<td>LKK</td>
<td>---</td>
<td>Variable used in the RPP section as an index to locate data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LN</td>
<td>---</td>
<td>Pointer to the location of the direction cosines of the normal for the TOR and TEC</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Location of the pointers in the LBODY section for the body being tested</td>
<td>ND</td>
</tr>
<tr>
<td>LOCARS</td>
<td>---</td>
<td>Beginning location of intersect data for the ARS</td>
<td>ND</td>
</tr>
<tr>
<td>LR1</td>
<td>---</td>
<td>Variable used in the REC, ELL, REC, TEC, and TOR sections as an index to locate data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>LR2</td>
<td>---</td>
<td>Variable used in the REC, ELL, and TEC sections as an index to locate data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LR3</td>
<td>---</td>
<td>Variable used in the TEC section as an index to locate data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LS</td>
<td>---</td>
<td>Variable used in the ELL section as an index to locate the length of the major axis from the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LSPT</td>
<td>---</td>
<td>Variable used in the ARB section to represent the location of the intersected plane data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LSURF</td>
<td>---</td>
<td>Surface number of the body where the intersect occurs (negative if an exit intersect)</td>
<td>ND</td>
</tr>
<tr>
<td>LV</td>
<td>---</td>
<td>Variable used as an index to locate the coordinates of the vertex in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LV1</td>
<td>---</td>
<td>Variable used as an index to locate the coordinates of the vertex or height vector in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LV2</td>
<td>---</td>
<td>Variable used as an index to locate the coordinates of the height vector in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LV3</td>
<td>---</td>
<td>Refer to LV1</td>
<td>ND</td>
</tr>
<tr>
<td>M</td>
<td>---</td>
<td>Variable used as an index to locate data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>MK</td>
<td>---</td>
<td>Variable used in the BOX section as an index to locate data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>NBO</td>
<td>---</td>
<td>Solid number of the body under test</td>
<td>ND</td>
</tr>
<tr>
<td>NEXREG</td>
<td>---</td>
<td>Region number of the region following the next intersect</td>
<td>ND</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>Variable used in section TEC to represent the length of the major radius of the base ellipse, and in section TOR to represent the major radius</td>
<td>Inches</td>
</tr>
<tr>
<td>R2</td>
<td>R2</td>
<td>Variable used in section TEC to represent the length of the minor radius of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>R4</td>
<td>R4</td>
<td>Variable used in section TEC to represent the length of the minor radius of the top ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>S1</td>
<td>---</td>
<td>Normal distance through the region that is returned by Subroutine GI</td>
<td>Inches</td>
</tr>
<tr>
<td>SUM</td>
<td>---</td>
<td>Variable used in sections REC and RAW to compute the dot product of two vectors, and in section ARB to compute the sum of the squares of the x, y, and z coefficients of the intersected plane</td>
<td>Inches²</td>
</tr>
<tr>
<td>TAU</td>
<td>(R1/R2)²</td>
<td>Variable used in section TEC to represent the square of the ratio of the semi-major axis radius to the semi-minor axis radius of the base ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>TEM(3)</td>
<td>---</td>
<td>Three-element array used to store the x, y, and z coordinates of a vector</td>
<td>Inches</td>
</tr>
<tr>
<td>TEM1(3)</td>
<td>---</td>
<td>Refer to TEM(3)</td>
<td>Inches</td>
</tr>
<tr>
<td>TEMP(3)</td>
<td>---</td>
<td>Refer to TEM(3)</td>
<td>Inches</td>
</tr>
<tr>
<td>TEM1P(3)</td>
<td>---</td>
<td>Refer to TEM(3)</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

#### SUBROUTINE CALC (Concluded)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLK</td>
<td>n</td>
<td>Variable used in the RAW section to compute and represent the scalar length of a vector normal to the slanted side of the RAW</td>
<td>Inches</td>
</tr>
<tr>
<td>TWOA</td>
<td>2a</td>
<td>Variable used in the TEC section to represent the length of the intersection ellipse along the semi-major axis</td>
<td>Inches</td>
</tr>
<tr>
<td>VF(3)</td>
<td>---</td>
<td>Array for storing the vertex coordinates of the TEC</td>
<td>Inches</td>
</tr>
<tr>
<td>WI(3)</td>
<td>WI</td>
<td>Coordinates of the direction cosines of the height vector of the RCC</td>
<td>Inches</td>
</tr>
<tr>
<td>WN(3)</td>
<td>WN</td>
<td>Three-element array used to store the x, y, and z coordinates of a unit vector</td>
<td>Inches</td>
</tr>
<tr>
<td>XI(3)</td>
<td>X</td>
<td>Three-element array used to store the x, y, and z coordinates of the intersect point</td>
<td>Inches</td>
</tr>
<tr>
<td>XMID(3)</td>
<td>(H1-H2)</td>
<td>Three-element array used in the RAW section to represent the x, y, and z coordinates of the H1-H2 vector</td>
<td>Inches</td>
</tr>
<tr>
<td>XNOS</td>
<td>Q</td>
<td>Variable used to represent a constant multiplier of the direction cosines of a ray and has a value of +1 or -1 and is used to direct the normal to a surface into the body for an entry intersect and away from the body for an exit intersect</td>
<td>ND</td>
</tr>
<tr>
<td>XP(3)</td>
<td>XH</td>
<td>Three-element array used in the RCC section to represent the x, y, and z coordinates of the intersect point projected onto the height vector</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index for referencing the temporary working storage section of Subroutine GI (LIO)</td>
<td>ND</td>
</tr>
<tr>
<td>I1</td>
<td>---</td>
<td>Entering surface number of an intersect for a given body from LIO</td>
<td>ND</td>
</tr>
<tr>
<td>I2</td>
<td>---</td>
<td>Exit surface number of an intersect for a given body from LIO</td>
<td>ND</td>
</tr>
<tr>
<td>I3</td>
<td>---</td>
<td>Equivalent of LOOP for a given body from LIO. Not used in Subroutine GI</td>
<td>ND</td>
</tr>
<tr>
<td>ICODE</td>
<td>---</td>
<td>Item code of the region before the present intersect</td>
<td>ND</td>
</tr>
<tr>
<td>ICODEI</td>
<td>---</td>
<td>Item code of the region after the present intersect</td>
<td>ND</td>
</tr>
<tr>
<td>IDENT</td>
<td>---</td>
<td>Space code of the region before the present intersect</td>
<td>ND</td>
</tr>
<tr>
<td>IDENTI</td>
<td>---</td>
<td>Space code of the region after the present intersect</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>---</td>
<td>Horizontal grid cell of the ray (from the center cell)</td>
<td>ND</td>
</tr>
<tr>
<td>IJK</td>
<td>---</td>
<td>Pointer for storing or locating data in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IRP</td>
<td>---</td>
<td>Number of an abutting RPP</td>
<td>ND</td>
</tr>
<tr>
<td>IRPRIM</td>
<td>---</td>
<td>Number of the next region</td>
<td>ND</td>
</tr>
<tr>
<td>ITEMP</td>
<td>---</td>
<td>Pointer to data in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>ITY</td>
<td>---</td>
<td>Body type number (1-12) of the current body</td>
<td>ND</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
### (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ITYPE</td>
<td>---</td>
<td>Unpacked body type number (1-11) of the current body</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Vertical grid cell of the ray (from the center cell)</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Pointer to the possible region being entered by the ray</td>
<td>ND</td>
</tr>
<tr>
<td>J1</td>
<td>---</td>
<td>Pointer to the first region in the enter (or leave) table for the current body</td>
<td>ND</td>
</tr>
<tr>
<td>J2</td>
<td>---</td>
<td>Pointer to the last region in the enter (or leave) table for the current body</td>
<td>ND</td>
</tr>
<tr>
<td>LEAV</td>
<td>---</td>
<td>Pointer to the region leaving table for the current body</td>
<td>ND</td>
</tr>
<tr>
<td>LENT</td>
<td>---</td>
<td>Pointer to the region entering table for the current body</td>
<td>ND</td>
</tr>
<tr>
<td>LION</td>
<td>---</td>
<td>Pointer to the last location in LIO, the temporary working storage for Subroutine Gl</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Pointer to data in the MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LSURT</td>
<td>---</td>
<td>Surface number for the current intersect of the current body (negative if an exit intersect)</td>
<td>ND</td>
</tr>
<tr>
<td>LTRUE</td>
<td>---</td>
<td>Indicator (0 or 1) to Subroutine Gl from Subroutine WOWI if point XP is in the region passed to Subroutine WOWI</td>
<td>ND</td>
</tr>
<tr>
<td>NASCT</td>
<td>---</td>
<td>Body number of the current intersect</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS

#### (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>NBO</td>
<td>---</td>
<td>Body number of the body under test</td>
<td>ND</td>
</tr>
<tr>
<td>NC</td>
<td>---</td>
<td>Number of bodies in the region description</td>
<td>ND</td>
</tr>
<tr>
<td>NEAV</td>
<td>---</td>
<td>Number of regions in the region entering table</td>
<td>ND</td>
</tr>
<tr>
<td>NHIT</td>
<td>---</td>
<td>Count of the number of bodies hit at a single intersect</td>
<td>ND</td>
</tr>
<tr>
<td>SI</td>
<td>---</td>
<td>Distance the ray has travelled into the region</td>
<td>Inches</td>
</tr>
<tr>
<td>SM</td>
<td>---</td>
<td>Distance to the next intersect of the body being tested that is greater than the distance travelled thus far</td>
<td>Inches</td>
</tr>
<tr>
<td>XBD(3)</td>
<td>---</td>
<td>x, y, and z coordinates of the ray when an error in Subroutine Gl occurred</td>
<td>Inches</td>
</tr>
<tr>
<td>XP(3)</td>
<td>---</td>
<td>x, y, and z coordinates of the position of the point on the ray</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>-----------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IJK</td>
<td>---</td>
<td>Pointer for locating and storing data in Subroutine Gl working storage, LIO</td>
<td>ND</td>
</tr>
<tr>
<td>IOP</td>
<td>---</td>
<td>Body operator of the body when unpacked from the region data section</td>
<td>ND</td>
</tr>
<tr>
<td>IOPER</td>
<td>---</td>
<td>Body operator of the body when unpacked from the region data section</td>
<td>ND</td>
</tr>
<tr>
<td>ITEMP</td>
<td>---</td>
<td>Pointer for locating body data pointers for the body being tested</td>
<td>ND</td>
</tr>
<tr>
<td>ITY</td>
<td>---</td>
<td>Body type number (1-12) of the current body</td>
<td>ND</td>
</tr>
<tr>
<td>LOCD</td>
<td>---</td>
<td>Pointer to the packed operator and body number for a specific body in the region under test</td>
<td>ND</td>
</tr>
<tr>
<td>LTRUE</td>
<td>---</td>
<td>Indicator (0 or 1) to Subroutine Gl if point XB is within the region</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Region body sequence number of the body being tested</td>
<td>ND</td>
</tr>
<tr>
<td>NBO</td>
<td>---</td>
<td>Body number of the body under test</td>
<td>ND</td>
</tr>
<tr>
<td>NC</td>
<td>---</td>
<td>Number of bodies in the region description</td>
<td>ND</td>
</tr>
<tr>
<td>NN</td>
<td>---</td>
<td>Region body sequence number of the body being compared</td>
<td>ND</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>---</td>
<td>DO loop index for representing one of six sides of the RPP</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>1</td>
<td>Index for denoting x, y, and z plane pairs</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Index that represents the x, y, or z coordinate of a point</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Index counter for the number of sides intersected</td>
<td>ND</td>
</tr>
<tr>
<td>LR(L)</td>
<td>---</td>
<td>Surface number of intersect point</td>
<td>ND</td>
</tr>
<tr>
<td>LST(1)</td>
<td>1</td>
<td>Integer that represents X plane</td>
<td>ND</td>
</tr>
<tr>
<td>LST(2)</td>
<td>1</td>
<td>Integer that represents X plane</td>
<td>ND</td>
</tr>
<tr>
<td>LST(3)</td>
<td>2</td>
<td>Integer that represents Y plane</td>
<td>ND</td>
</tr>
<tr>
<td>LST(4)</td>
<td>2</td>
<td>Integer that represents Y plane</td>
<td>ND</td>
</tr>
<tr>
<td>LST(5)</td>
<td>3</td>
<td>Integer that represents Z plane</td>
<td>ND</td>
</tr>
<tr>
<td>LST(6)</td>
<td>3</td>
<td>Integer that represents Z plane</td>
<td>ND</td>
</tr>
<tr>
<td>NBO</td>
<td>---</td>
<td>Body number of the body under test</td>
<td>ND</td>
</tr>
<tr>
<td>PR(L)</td>
<td>---</td>
<td>Location in subroutine for RIN and ROUT</td>
<td>Inches</td>
</tr>
<tr>
<td>TEMP</td>
<td>$\frac{XS_I - XB_J}{WB_J}$</td>
<td>Numerator of the equation for distance to the intersect point of ray with plane</td>
<td>Inches</td>
</tr>
<tr>
<td>TRY</td>
<td>$\frac{XS_I - XB_S}{WB_J}$</td>
<td>Distance from ray origin to intersect point with plane</td>
<td>Inches</td>
</tr>
<tr>
<td>XRY</td>
<td>$\frac{XB + WB \cdot S_I}{XS_I}$</td>
<td>Intersect coordinate on a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>XS(I)</td>
<td>$\frac{XS_I}{XS_I}$</td>
<td>Boundary coordinate for a given plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>A</td>
<td>---</td>
<td>Dot product $\vec{H}_1 \cdot \vec{H}_i$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>CM</td>
<td>RINI</td>
<td>Computed value for distance to the first intersection with the box, RIN</td>
<td>Inches</td>
</tr>
<tr>
<td>CP</td>
<td>ROUTI</td>
<td>Computed value for distance to the second intersection with the box, ROUT</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>DO loop index for computing the face pair</td>
<td>ND</td>
</tr>
<tr>
<td>IH1</td>
<td>---</td>
<td>Location in ASTER array which contains $\vec{H}_1$ coordinates</td>
<td>ND</td>
</tr>
<tr>
<td>IH2</td>
<td>---</td>
<td>Location in ASTER array which contains $\vec{H}_2$ coordinates</td>
<td>ND</td>
</tr>
<tr>
<td>IH3</td>
<td>---</td>
<td>Location in ASTER array which contains $\vec{H}_3$ coordinates</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>---</td>
<td>Integer used to compute the face number of the box</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Location in the ASTER array which contains the vertex coordinates</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Index for looping through the x, y, and z coordinates for computing the equation for the distance to the intersects</td>
<td>ND</td>
</tr>
<tr>
<td>JA</td>
<td>---</td>
<td>x, y, or z coordinate of the $\vec{H}_1$ vector</td>
<td>Inches</td>
</tr>
<tr>
<td>JV</td>
<td>---</td>
<td>x, y, or z coordinate of the vertex</td>
<td>Inches</td>
</tr>
<tr>
<td>LI</td>
<td>---</td>
<td>Face number for ray's first intercept</td>
<td>ND</td>
</tr>
<tr>
<td>LO</td>
<td>---</td>
<td>Face number for ray's second intercept</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Index used to retrieve pointers to the $H_1$ and $H_2$ coordinates</td>
<td>ND</td>
</tr>
<tr>
<td>VP</td>
<td>$(\vec{V} - \vec{XB}) \cdot \vec{H}$</td>
<td>Dot product $(\vec{V} - \vec{XB}) \cdot \vec{H}$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>W</td>
<td>$\vec{WB} \cdot \vec{H}$</td>
<td>Dot product $\vec{WB} \cdot \vec{H}$</td>
<td>Inches$^2$</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS

(SIMULATION MODEL)

## SUBROUTINE SPH

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>DX·WB</td>
<td>B coefficient of the quadratic equation solution used to compute RIN and ROUT</td>
<td>Inches</td>
</tr>
<tr>
<td>C</td>
<td>DX²-R²</td>
<td>C coefficient of the quadratic equation solution used to compute RIN and ROUT</td>
<td>Inches²</td>
</tr>
<tr>
<td>DIS</td>
<td>B²-C</td>
<td>Quantity under the radical of the quadratic equation solution used to compute RIN and ROUT</td>
<td>Inches²</td>
</tr>
<tr>
<td>DX, DY, DZ</td>
<td>---</td>
<td>x, y and z coordinates of the vector DX</td>
<td>Inches</td>
</tr>
<tr>
<td>I2</td>
<td>---</td>
<td>Pointer to the location of the radius in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>ITEMP</td>
<td>---</td>
<td>Location in ASTER array which contains the vertex coordinates</td>
<td>ND</td>
</tr>
<tr>
<td>R</td>
<td>R</td>
<td>Radius of sphere</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### SUBROUTINE RCC

**LIST OF SYMBOLS AND ABBREVIATIONS**  
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AMBD</td>
<td>$\lambda'$</td>
<td>Value of $\lambda'$ from the equation $\tau S^2 - 2S\lambda' + \mu' = 0$</td>
<td>Inches</td>
</tr>
<tr>
<td>AMBDA</td>
<td>$\lambda$</td>
<td>Coefficient of $2S$ from the quadratic equation of the RCC</td>
<td>Inches</td>
</tr>
<tr>
<td>CM</td>
<td>---</td>
<td>RIN intersection with a planar surface</td>
<td>Inches</td>
</tr>
<tr>
<td>CP</td>
<td>---</td>
<td>ROUT intersection with a planar surface</td>
<td>Inches</td>
</tr>
<tr>
<td>DEN</td>
<td>$\tau$</td>
<td>Coefficient of $S^2$ from the quadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>DISC</td>
<td>$\lambda^2 - \mu$</td>
<td>Quantity under the radical of the quadratic equation used to compute RIN and ROUT</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>Fl</td>
<td>---</td>
<td>A quantity used to determine if RIN or ROUT lies within the boundary of the RCC</td>
<td>Inches</td>
</tr>
<tr>
<td>H(3)</td>
<td>$\vec{H}$</td>
<td>$x$, $y$, and $z$ coordinates of the height vector $\vec{H}$</td>
<td>Inches</td>
</tr>
<tr>
<td>HH</td>
<td>$\vec{H} \cdot \vec{H}$</td>
<td>Dot product of the height vector $\vec{H}$ of the RCC</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index that represents an $x$, $y$, or $z$ coordinate</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>---</td>
<td>$x$, $y$, and $z$ coordinates of the light vector $\vec{H}$</td>
<td>Inches</td>
</tr>
<tr>
<td>IRR</td>
<td>---</td>
<td>Location index for the radius $R$ of the RCC in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>$x$, $y$, and $z$ coordinates of the vertex $\vec{V}$</td>
<td>Inches</td>
</tr>
<tr>
<td>LCM</td>
<td>---</td>
<td>Temporary storage location for the surface number of RIN for a planar surface</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>LCP</td>
<td></td>
<td>Temporary storage location for the surface number of ROUT for a planar surface</td>
<td>ND</td>
</tr>
<tr>
<td>POT</td>
<td>$(XB-V)^2$</td>
<td>Portion of the expression for solving for $\mu$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>R</td>
<td>$R$</td>
<td>Radius of the right circular cylinder</td>
<td>Inches</td>
</tr>
<tr>
<td>R1</td>
<td>$\lambda - \sqrt{\lambda^2 - \mu}$</td>
<td>Temporary location in the subroutine of RIN for the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>R2</td>
<td>$\lambda + \sqrt{\lambda^2 - \mu}$</td>
<td>Temporary location in the subroutine of ROUT for the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>RSQ</td>
<td>$R^2$</td>
<td>Value of radius squared</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>SD</td>
<td>$\sqrt{\lambda^2 - \mu}$</td>
<td>Value $\sqrt{\lambda^2 - \mu}$ from the expression $S = \lambda \pm \sqrt{\lambda^2 - \mu}$</td>
<td>Inches</td>
</tr>
<tr>
<td>TOP</td>
<td>$\overline{WB} \cdot (XB-V)$</td>
<td>Portion of the expression for solving for $\lambda$</td>
<td>Inches</td>
</tr>
<tr>
<td>UM</td>
<td>$\mu'$</td>
<td>Value of $\mu'$ from the quadratic equation $ts^2 - 2s\lambda' + \mu' = 0$</td>
<td>Inches</td>
</tr>
<tr>
<td>UMU</td>
<td>$\mu$</td>
<td>$\mu = \mu' / \tau$</td>
<td>Inches</td>
</tr>
<tr>
<td>V(3)</td>
<td>$V$</td>
<td>$x$, $y$, and $z$ coordinates of the vertex $V$</td>
<td>Inches</td>
</tr>
<tr>
<td>VPH</td>
<td>$\overline{H} \cdot (V-\overline{XB})$</td>
<td>Quantity used in solving for $\lambda$ and $\mu$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>WH</td>
<td>$\overline{WB} \cdot \overline{H}$</td>
<td>Quantity used in solving for $\gamma$ and $\lambda$ and RIN and ROUT on the planar surfaces</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(3)</td>
<td>( \overline{A} )</td>
<td>( x, y, ) and ( z ) coordinates of the semi-major axis ( A )</td>
<td>Inches</td>
</tr>
<tr>
<td>AA</td>
<td>( \overline{A} \cdot \overline{A} )</td>
<td>Dot product of the semi-major axis ( A )</td>
<td>Inches²</td>
</tr>
<tr>
<td>AAAA</td>
<td>((\overline{A} \cdot \overline{A})^2)</td>
<td>Square of the dot product ( A \cdot A )</td>
<td>Inches⁴</td>
</tr>
<tr>
<td>AMBD</td>
<td>( \lambda' )</td>
<td>Coefficient of 2S from equation ( \tau S^2 + 2\lambda' S + \mu' = 0 )</td>
<td>ND</td>
</tr>
<tr>
<td>AMBDA</td>
<td>( \lambda'/\tau )</td>
<td>Coefficient, ( \lambda ), of 2S from equation ( S^2 + 2\lambda S + \mu = 0 )</td>
<td>ND</td>
</tr>
<tr>
<td>B(3)</td>
<td>( \overline{B} )</td>
<td>( x, y, ) and ( z ) coordinates of the semi-minor axis ( B )</td>
<td>Inches</td>
</tr>
<tr>
<td>BB</td>
<td>( \overline{B} \cdot \overline{B} )</td>
<td>Dot product of the semi-minor axis ( B )</td>
<td>Inches²</td>
</tr>
<tr>
<td>BBBB</td>
<td>((\overline{B} \cdot \overline{B})^2)</td>
<td>Square of the dot product ( B \cdot B )</td>
<td>Inches⁴</td>
</tr>
<tr>
<td>CM</td>
<td>---</td>
<td>RIN intersection with a planar surface</td>
<td>Inches</td>
</tr>
<tr>
<td>CP</td>
<td>---</td>
<td>ROUT intersection with a planar surface</td>
<td>Inches</td>
</tr>
<tr>
<td>DEN</td>
<td>( \tau )</td>
<td>Coefficient of ( S^2 ) from the quadratic equation of the REC</td>
<td>ND</td>
</tr>
<tr>
<td>DISC</td>
<td>( \lambda^2 - \mu )</td>
<td>Quantity under the radical of the quadratic equation used in solving for RIN and ROUT on the quadratic surface</td>
<td>Inches²</td>
</tr>
<tr>
<td>FL</td>
<td>---</td>
<td>Quantity used to determine if RIN or ROUT lies within the boundary of the REC</td>
<td>Inches</td>
</tr>
<tr>
<td>H(3)</td>
<td>( \overline{H} )</td>
<td>( x, y, ) and ( z ) coordinates of the height vector ( H )</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS

### SUBROUTINE REC (Continued)  (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>HH</td>
<td>$\bar{H} \cdot \bar{H}$</td>
<td>Dot product, $\bar{H} \cdot \bar{H}$, of the height vector $\bar{H}$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>IA</td>
<td>---</td>
<td>Pointer to the coordinates of the semi-major axis $\bar{A}$</td>
<td>ND</td>
</tr>
<tr>
<td>IB</td>
<td>---</td>
<td>Pointer to the coordinates of the semi-minor axis $\bar{B}$</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>---</td>
<td>Pointer to the coordinates of the height vector $\bar{H}$</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Pointer to the coordinates of the vertex $\bar{V}$</td>
<td>ND</td>
</tr>
<tr>
<td>LCM</td>
<td>---</td>
<td>Temporary storage location for the surface number of RIN for a planar surface</td>
<td>ND</td>
</tr>
<tr>
<td>LCP</td>
<td>---</td>
<td>Temporary storage location for the surface number of ROUT for a planar surface</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Index used to retrieve pointers to the location of the coordinates of the REC</td>
<td>ND</td>
</tr>
<tr>
<td>R1</td>
<td>$\lambda - \sqrt{\lambda^2 - \mu}$</td>
<td>Temporary storage location of RIN for the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>R2</td>
<td>$\lambda + \sqrt{\lambda^2 - \mu}$</td>
<td>Temporary storage location of ROUT for the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>SD</td>
<td>$\sqrt{\lambda^2 - \mu}$</td>
<td>Value $\sqrt{\lambda^2 - \mu}$ from the expression $s = \lambda + \sqrt{\lambda^2 - \mu}$</td>
<td>Inches</td>
</tr>
<tr>
<td>UM</td>
<td>$\mu'$</td>
<td>Value of $\mu'$ from the quadratic equation $\tau s^2 - 2s\lambda' + \mu' = 0$</td>
<td>ND</td>
</tr>
<tr>
<td>UMU</td>
<td>$\mu$</td>
<td>$\mu = \mu'/\tau$</td>
<td>ND</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS
## SUBROUTINE REC (Concluded) (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>V(3)</td>
<td>$\bar{V}$</td>
<td>$x$, $y$, and $z$ coordinates of the vertex $V$</td>
<td>Inches</td>
</tr>
<tr>
<td>V1XB1</td>
<td>$\bar{V}_x - \bar{X}_B$</td>
<td>$x$ component of the vector $\bar{V} - \bar{X}_B$</td>
<td>Inches</td>
</tr>
<tr>
<td>V2XB2</td>
<td>$\bar{V}_y - \bar{X}_B$</td>
<td>$y$ component of the vector $\bar{V} - \bar{X}_B$</td>
<td>Inches</td>
</tr>
<tr>
<td>V3XB3</td>
<td>$\bar{V}_z - \bar{X}_B$</td>
<td>$z$ component of the vector $\bar{V} - \bar{X}_B$</td>
<td>Inches</td>
</tr>
<tr>
<td>VPA</td>
<td>$\bar{A} \cdot (\bar{V} - \bar{X}_B)$</td>
<td>Dot product of the semi-major axis $\bar{A}$ and the vector from $\bar{X}_B$ to the vertex of the REC $(\bar{V} - \bar{X}_B)$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>VPB</td>
<td>$\bar{B} \cdot (\bar{V} - \bar{X}_B)$</td>
<td>Dot product of the semi-minor axis $\bar{B}$ and the vector from $\bar{X}_B$ to the vertex of the REC $(\bar{V} - \bar{X}_B)$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>VPH</td>
<td>$\bar{H} \cdot (\bar{V} - \bar{X}_B)$</td>
<td>Dot product of the height vector $\bar{H}$ and the vector from $\bar{X}_B$ to the vertex of the REC $(\bar{V} - \bar{X}_B)$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>VPHHH</td>
<td>$\bar{H} \cdot (\bar{V} - \bar{X}_B) + \bar{H} \cdot \bar{H}$</td>
<td>Sum of the dot product $\bar{H} \cdot (\bar{V} - \bar{X}_B)$ and the dot product of the height vector $\bar{H} \cdot \bar{H}$</td>
<td>Inches</td>
</tr>
<tr>
<td>WBA</td>
<td>$\bar{W}_B \cdot \bar{A}$</td>
<td>Dot product of the semi-major axis $\bar{A}$ and the direction cosines of the ray $\bar{W}_B$</td>
<td>Inches</td>
</tr>
<tr>
<td>WBAWBA</td>
<td>$(\bar{W}_B \cdot \bar{A})^2$</td>
<td>Square of the dot product $\bar{W}_B \cdot \bar{A}$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>WBB</td>
<td>$\bar{W}_B \cdot \bar{B}$</td>
<td>Dot product of the semi-minor axis $\bar{B}$ and the direction cosines of the ray $\bar{W}_B$</td>
<td>Inches</td>
</tr>
<tr>
<td>WBBWBB</td>
<td>$(\bar{W}_B \cdot \bar{B})^2$</td>
<td>Square of the dot product $\bar{W}_B \cdot \bar{B}$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>WH</td>
<td>$\bar{W}_B \cdot \bar{H}$</td>
<td>Quantity used for determining the direction of the ray with respect to the planar surfaces, and it is used in solving for RIN and ROUT of the planar surfaces</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>AMBD</td>
<td>$\lambda'$</td>
<td>Coefficient, $\lambda'$, of 2S from equation $\tau S^2 - 2\lambda' S + \mu' = 0$</td>
<td>ND</td>
</tr>
<tr>
<td>AMBDIA</td>
<td>$\lambda'/\tau$</td>
<td>Coefficient, $\lambda'$, of 2S from equation $S^2 - 2\lambda S + \mu = 0$</td>
<td>Inches</td>
</tr>
<tr>
<td>CM</td>
<td>--</td>
<td>Temporary storage location of RIN for a planar surface</td>
<td>Inches</td>
</tr>
<tr>
<td>CP</td>
<td>--</td>
<td>Temporary storage location of ROUT for a planar surface</td>
<td>Inches</td>
</tr>
<tr>
<td>DEN</td>
<td>$\tau$</td>
<td>Coefficient, $\tau$, of $S^2$ from equation $\tau S^2 - 2\lambda' S + \mu' = 0$</td>
<td>ND</td>
</tr>
<tr>
<td>DISC</td>
<td>$\lambda^2 - \mu$</td>
<td>Quantity under the radical of the quadratic equation used in solving for RIN and ROUT on the quadratic surface</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>Fl</td>
<td>--</td>
<td>Variable used to determine if an intersect lies within the boundaries of the TRC</td>
<td>Inches</td>
</tr>
<tr>
<td>H(3)</td>
<td>$\bar{H}$</td>
<td>x, y, and z coordinates of height vector $\bar{H}$</td>
<td>Inches</td>
</tr>
<tr>
<td>HH</td>
<td>$\bar{H} \cdot \bar{H}$</td>
<td>Dot product of height vector ($\bar{H} \cdot \bar{H}$)</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>IH</td>
<td>--</td>
<td>Pointer to the coordinates of the height vector</td>
<td>ND</td>
</tr>
<tr>
<td>INTR1</td>
<td>--</td>
<td>Counter for counting the number of intersects for RIN</td>
<td>ND</td>
</tr>
<tr>
<td>INTR2</td>
<td>--</td>
<td>Counter for counting the number of intersects for ROUT</td>
<td>ND</td>
</tr>
<tr>
<td>INTSEC</td>
<td>--</td>
<td>Counter for counting the number of valid intersects</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IRB</td>
<td>--</td>
<td>Pointer to the value of the radius of the base</td>
<td>ND</td>
</tr>
<tr>
<td>IRTOP</td>
<td>--</td>
<td>Pointer to the value of the radius of the top</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>--</td>
<td>Pointer to the coordinates of the vertex</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>--</td>
<td>Location of the pointers to the radii of the top and bottom of the TRC</td>
<td>ND</td>
</tr>
<tr>
<td>PVPV</td>
<td>$(V-XB)^2$</td>
<td>Dot product of vector $(V-XB)$ $(V-XB)$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>R1</td>
<td>$\lambda-\sqrt{\lambda^2-\mu}$</td>
<td>Temporary storage location of RIN or ROUT for the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>R2</td>
<td>$\mu'/2\lambda'$ or $\frac{\lambda+\sqrt{\lambda^2-\mu}}{2}$</td>
<td>Temporary storage location of RIN or ROUT for the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>RB</td>
<td>$R_B$</td>
<td>Radius at vertex $V$</td>
<td>Inches</td>
</tr>
<tr>
<td>RBRTVP</td>
<td>$C_2$</td>
<td>Portion of the constant term used in solving the quadratic equation</td>
<td>Inches</td>
</tr>
<tr>
<td>RT</td>
<td>$R_T$</td>
<td>Radius at $V+H$</td>
<td>Inches</td>
</tr>
<tr>
<td>RTRB</td>
<td>$R_T-R_B$</td>
<td>Difference between the radius of the upper base radius and the radius of the lower base</td>
<td>Inches</td>
</tr>
<tr>
<td>SD</td>
<td>$\sqrt{\lambda^2-\mu}$</td>
<td>Value $\sqrt{\lambda^2-\mu}$ from the expression $S = \lambda+\sqrt{\lambda^2-\mu}$</td>
<td>Inches</td>
</tr>
<tr>
<td>UM</td>
<td>$\mu'$</td>
<td>Value of the constant term from the quadratic equation $\tau S^2-2\lambda'S+\mu'=0$</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>UMU</td>
<td>$\mu$</td>
<td>$\mu = \mu' / \tau$ from the equation $S^2 - 2\lambda S + \mu = 0$</td>
<td>ND</td>
</tr>
<tr>
<td>V(3)</td>
<td>$\overline{V}$</td>
<td>$x, y,$ and $z$ coordinate of the vertex $V$</td>
<td>Inches</td>
</tr>
<tr>
<td>V1XB1</td>
<td>$\overline{V} - \overline{XB}_x$</td>
<td>$x$ component of vector $(\overline{V} - \overline{XB})$</td>
<td>Inches</td>
</tr>
<tr>
<td>V2XB2</td>
<td>$\overline{V} - \overline{XB}_y$</td>
<td>$y$ component of vector $(\overline{V} - \overline{XB})$</td>
<td>Inches</td>
</tr>
<tr>
<td>V3XB3</td>
<td>$\overline{V} - \overline{XB}_z$</td>
<td>$z$ component of vector $(\overline{V} - \overline{XB})$</td>
<td>Inches</td>
</tr>
<tr>
<td>VPH</td>
<td>$(\overline{V} - \overline{XB}) \cdot \overline{H}$</td>
<td>Dot product of vector $(\overline{V} - \overline{XB})$ and height vector $\overline{H}$</td>
<td>Inches²</td>
</tr>
<tr>
<td>VPHHH</td>
<td>$(\overline{V} - \overline{XB}) \cdot \overline{H} + \overline{H} \cdot \overline{H}$</td>
<td>Sum of dot product of vector $(\overline{V} - \overline{XB})$ and height vector $\overline{H}$ plus dot product of height vector $\overline{H}$</td>
<td>Inches²</td>
</tr>
<tr>
<td>VPW</td>
<td>$(\overline{V} - \overline{XB}) \cdot \overline{WB}$</td>
<td>Dot product of vector $(\overline{V} - \overline{XB})$ and direction cosines of the ray</td>
<td>Inches</td>
</tr>
<tr>
<td>WH</td>
<td>$\overline{WB} \cdot \overline{H}$</td>
<td>Dot product of direction cosines of the ray and height vector $\overline{H}$</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
**(SIMULATION MODEL)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>$2 \overrightarrow{D_1} \cdot \overrightarrow{WB}$</td>
<td>Vector dot product $2 \overrightarrow{D_1} \cdot \overrightarrow{WB}$</td>
<td>Inches</td>
</tr>
<tr>
<td>A2</td>
<td>$2 \overrightarrow{D_2} \cdot \overrightarrow{WB}$</td>
<td>Vector dot product $2 \overrightarrow{D_2} \cdot \overrightarrow{WB}$</td>
<td>Inches</td>
</tr>
<tr>
<td>AA</td>
<td>B</td>
<td>Second term of Equation (64)</td>
<td>ND</td>
</tr>
<tr>
<td>ALAM1</td>
<td>$\lambda$</td>
<td>Coefficient of quadratic equation used to compute RIN and ROUT</td>
<td>ND</td>
</tr>
<tr>
<td>ALAMD</td>
<td>$B^2-1$</td>
<td>Intermediate variable</td>
<td>ND</td>
</tr>
<tr>
<td>B1</td>
<td>$(D_1)^2$</td>
<td>Vector dot product $\overrightarrow{D_1}^2$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>B2</td>
<td>$(D_2)^2$</td>
<td>Vector dot product $\overrightarrow{D_2}^2$</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>BB</td>
<td>A</td>
<td>First term of Equation (64)</td>
<td>ND</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>Length of ellipsoid major axis</td>
<td>Inches</td>
</tr>
<tr>
<td>D1X, D1Y, D1Z</td>
<td>DL</td>
<td>x, y, and z coordinates of $\overrightarrow{XB}$ - (Foci A)</td>
<td>Inches</td>
</tr>
<tr>
<td>D2X, D2Y, D2Z</td>
<td>D2</td>
<td>x, y, and z coordinates of $\overrightarrow{XB}$ - (Foci B)</td>
<td>Inches</td>
</tr>
<tr>
<td>DISCRM</td>
<td>$\lambda^2 - \mu$</td>
<td>Quantity under radical of quadratic equation solution</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>FOCIA(3)</td>
<td>$\overrightarrow{F_a}$</td>
<td>x, y, and z coordinates of one of the ellipsoid's foci</td>
<td>Inches</td>
</tr>
<tr>
<td>FOCIB(3)</td>
<td>$\overrightarrow{F_b}$</td>
<td>x, y, and z coordinates of one of the ellipsoid's foci</td>
<td>Inches</td>
</tr>
<tr>
<td>IRR</td>
<td>---</td>
<td>Location where the ellipsoid's major axis length is stored</td>
<td>ND</td>
</tr>
<tr>
<td>IV1</td>
<td>---</td>
<td>Location of the coordinates of one of the ellipsoid's foci</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS

**SUBROUTINE ELL (Concluded) (SIMULATION MODEL)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV2</td>
<td>---</td>
<td>Location of the coordinates of the other ellipsoid's foci</td>
<td>ND</td>
</tr>
<tr>
<td>SQRTDI</td>
<td>(\sqrt{\lambda^2 - \mu})</td>
<td>Square root of ((\lambda^2 - \mu)) for solving for RIN and ROUT</td>
<td>Inches</td>
</tr>
<tr>
<td>U</td>
<td>(\mu)</td>
<td>Coefficient of quadratic equation used to compute RIN and ROUT</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td>AG</td>
<td>A&lt;sub&gt;2&lt;/sub&gt;G&lt;sub&gt;1&lt;/sub&gt; + A&lt;sub&gt;1&lt;/sub&gt;G&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Denominator for computing S&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Inches&lt;sup&gt;3&lt;/sup&gt;</td>
</tr>
<tr>
<td>ASQ(1)</td>
<td>A&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Dot product H&lt;sub&gt;1&lt;/sub&gt; · H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>Inches&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>ASQ(2)</td>
<td>A&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Dot product H&lt;sub&gt;2&lt;/sub&gt; · H&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Inches&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>ASQ(3)</td>
<td>A&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Dot product H&lt;sub&gt;3&lt;/sub&gt; · H&lt;sub&gt;3&lt;/sub&gt;</td>
<td>Inches&lt;sup&gt;2&lt;/sup&gt;</td>
</tr>
<tr>
<td>CM</td>
<td>---</td>
<td>Intermediate storage location for RIN</td>
<td>Inches</td>
</tr>
<tr>
<td>CP</td>
<td>---</td>
<td>Intermediate storage location for ROUT</td>
<td>Inches</td>
</tr>
<tr>
<td>G(1)</td>
<td>G&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Dot product WB · H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>Inches</td>
</tr>
<tr>
<td>G(2)</td>
<td>G&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Dot product WB · H&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Inches</td>
</tr>
<tr>
<td>G(3)</td>
<td>G&lt;sub&gt;i&lt;/sub&gt;</td>
<td>Dot product WB · H&lt;sub&gt;3&lt;/sub&gt;</td>
<td>Inches</td>
</tr>
<tr>
<td>H&lt;sub&gt;1&lt;/sub&gt;(3)</td>
<td>H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>x, y and z coordinates of vector H&lt;sub&gt;1&lt;/sub&gt;</td>
<td>Inches</td>
</tr>
<tr>
<td>H&lt;sub&gt;2&lt;/sub&gt;(3)</td>
<td>H&lt;sub&gt;2&lt;/sub&gt;</td>
<td>x, y and z coordinates of vector H&lt;sub&gt;2&lt;/sub&gt;</td>
<td>Inches</td>
</tr>
<tr>
<td>H&lt;sub&gt;3&lt;/sub&gt;(3)</td>
<td>H&lt;sub&gt;3&lt;/sub&gt;</td>
<td>x, y and z coordinates of the vector H&lt;sub&gt;3&lt;/sub&gt;</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index used to determine if intersections with sides 1 and 3 are possible</td>
<td>ND</td>
</tr>
<tr>
<td>IH&lt;sub&gt;1&lt;/sub&gt;</td>
<td>---</td>
<td>Pointer to the coordinates of the H&lt;sub&gt;1&lt;/sub&gt; vector</td>
<td>ND</td>
</tr>
<tr>
<td>IH&lt;sub&gt;2&lt;/sub&gt;</td>
<td>---</td>
<td>Pointer to the coordinates of the H&lt;sub&gt;2&lt;/sub&gt; vector</td>
<td>ND</td>
</tr>
<tr>
<td>IH&lt;sub&gt;3&lt;/sub&gt;</td>
<td>---</td>
<td>Pointer to the coordinates of the H&lt;sub&gt;3&lt;/sub&gt; vector</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS

#### SUBROUTINE RAW (Concluded) (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IV</td>
<td>---</td>
<td>Pointer to the coordinates of the vertex</td>
<td>ND</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>Variable used to determine if variables CP and CM have been updated with a computed intersection distance</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Variable used to determine if variables CP and CM have been updated with the computed intersection distance</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Index used to retrieve pointers to the location of the coordinates of the RAW</td>
<td>ND</td>
</tr>
<tr>
<td>PV(1)</td>
<td>Pi</td>
<td>Dot product ((\overline{XB} - \overline{V}) \cdot H_1)</td>
<td>Inches²</td>
</tr>
<tr>
<td>PV(2)</td>
<td>Pi</td>
<td>Dot product ((\overline{XB} - \overline{V}) \cdot H_2)</td>
<td>Inches²</td>
</tr>
<tr>
<td>PV(3)</td>
<td>Pi</td>
<td>Dot product ((\overline{XB} - \overline{V}) \cdot H_3)</td>
<td>Inches²</td>
</tr>
<tr>
<td>PV(4)</td>
<td>-(P1A2+P2A1)</td>
<td>Part of the numerator for computing S2</td>
<td>ND</td>
</tr>
<tr>
<td>TEMP</td>
<td>---</td>
<td>Intermediate variable used as temporary storage</td>
<td>Inches</td>
</tr>
<tr>
<td>TOP</td>
<td>A1A2-P1A2-P2A1</td>
<td>Numerator for computing S2</td>
<td>ND</td>
</tr>
<tr>
<td>V(3)</td>
<td>(\overline{V})</td>
<td>x, y and z coordinates of the vector (\overline{V})</td>
<td>Inches</td>
</tr>
<tr>
<td>XB1V1, XB2V2, XB3V3</td>
<td>(\overline{XB}-\overline{V})</td>
<td>x, y and z coordinates of the vector difference (\overline{XB} - \overline{V})</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
#### (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>AA (I,1)</td>
<td>Ai</td>
<td>Coefficients of the equation for the ith face of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>AA (I,2)</td>
<td>Bi</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td>AA (I,3)</td>
<td>Ci</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td>AA (I,4)</td>
<td>Di</td>
<td>ND</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>Di</td>
<td>The D coefficient of the ith plane equation</td>
<td>ND</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>The ith face of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>The jth face of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>K</td>
<td>---</td>
<td>The kth coordinate (x, y, or z)</td>
<td>ND</td>
</tr>
<tr>
<td>L1</td>
<td>---</td>
<td>Intermediate variable used to store the face number containing the first computed intersection</td>
<td>ND</td>
</tr>
<tr>
<td>L2</td>
<td>---</td>
<td>Intermediate variable used to store the face number containing the second computed intersection</td>
<td>ND</td>
</tr>
<tr>
<td>LC</td>
<td>---</td>
<td>Pointer to the coefficients of the ith plane</td>
<td>ND</td>
</tr>
<tr>
<td>LD</td>
<td>---</td>
<td>Pointer to the constant term of the ith plane</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Index used to retrieve pointers to the ARB constants</td>
<td>ND</td>
</tr>
<tr>
<td>S</td>
<td>Si</td>
<td>Distance to the intersect for the ith face of the ARB</td>
<td>Inches</td>
</tr>
<tr>
<td>S1</td>
<td>---</td>
<td>Distance for the first intersect computed</td>
<td>Inches</td>
</tr>
<tr>
<td>S2</td>
<td>---</td>
<td>Distance for the second intersect computed</td>
<td>Inches</td>
</tr>
<tr>
<td>SDEN</td>
<td>---</td>
<td>Denominator of the equation for computing the distance to the ith face of the ARB</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNUM</td>
<td>---</td>
<td>Numerator of the equation for computing the distance to the ith face of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>T</td>
<td>---</td>
<td>Intermediate variable used to test the equation for computing the distance to the ith face of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>T1</td>
<td>---</td>
<td>Intermediate variable used to test the equation for computing the distance to the ith face of the ARB</td>
<td>ND</td>
</tr>
<tr>
<td>XP</td>
<td>XI1</td>
<td>x, y, and z coordinates of the point of intersection on the ith plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>A</td>
<td>$\gamma R_3 + R_1 (1-\gamma)$</td>
<td>Semi-major axis of the intersection ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>AA(3)</td>
<td>$\overline{AA}$</td>
<td>$x$, $y$, and $z$ components of the semi-major axis unit vector of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>ALPHA</td>
<td>$\frac{\overline{H \cdot N}}{WB \cdot N}$</td>
<td>Distance along the ray from the plane of the base ellipse to the plane of the top ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>AMBDA</td>
<td>$\lambda$</td>
<td>Coefficient of $2S$ in the equation $TS^2 + 2AS + \mu = 0$</td>
<td>ND</td>
</tr>
<tr>
<td>ASQ</td>
<td>$(\gamma R_3 + R_1 (1-\gamma))^2$</td>
<td>Square of the semi-major axis of the intersection ellipse</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>B</td>
<td>$\gamma R_4 + R_2 (1-\gamma)$</td>
<td>Semi-minor axis of the intersection ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>BB(3)</td>
<td>$\overline{BB}$</td>
<td>$x$, $y$, and $z$ coordinates of the semi-minor axis unit vector of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>BETA</td>
<td>$\frac{(V-XB) \cdot \overline{N}}{WB \cdot N}$</td>
<td>Distance along the ray from start point XB to plane of base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>BSQ</td>
<td>$(\gamma R_4 + R_2 (1-\gamma))^2$</td>
<td>Square of the semi-minor axis of the intersection ellipse</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>DEN</td>
<td>$\tau$</td>
<td>Coefficient of $S^2$ in the equation $TS^2 + 2AS + \mu = 0$</td>
<td>ND</td>
</tr>
<tr>
<td>DISC</td>
<td>$\lambda^2 - \tau \mu$</td>
<td>Value under the radical of the quadratic equation used in solving for RIN and ROUT on the quadratic surface</td>
<td>ND</td>
</tr>
<tr>
<td>F</td>
<td>---</td>
<td>Intermediate variable used for determining if an intersect either lies between the two planes, or lies within the ellipse of a plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>F1</td>
<td>---</td>
<td>Distance from the center of the top or base ellipse to the intersection along the ( \bar{A} ) axis</td>
<td>Inches</td>
</tr>
<tr>
<td>F2</td>
<td>---</td>
<td>Distance from the center of the top or base ellipse to the intersection along the ( \bar{B} ) axis</td>
<td>Inches</td>
</tr>
<tr>
<td>GAMMA</td>
<td>( \frac{(V-XB) \cdot \bar{N}}{H \cdot \bar{N}} )</td>
<td>Ratio on normal of height of hit ( x, y, ) and ( z ) components of the height vector ( H )</td>
<td>ND</td>
</tr>
<tr>
<td>H(3)</td>
<td>( \bar{H} )</td>
<td>( x, y, ) and ( z ) components of the height vector ( H )</td>
<td>Inches</td>
</tr>
<tr>
<td>HDA</td>
<td>( \bar{H} \cdot \bar{A} )</td>
<td>Dot product of the height vector ( H ) and the semi-major axis unit vector ( A ) of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>HDB</td>
<td>( \bar{H} \cdot \bar{B} )</td>
<td>Dot product of the height vector ( H ) and the semi-minor axis unit vector ( B ) of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>HDN</td>
<td>( \bar{H} \cdot \bar{N} )</td>
<td>Dot product of the height vector ( H ) and the normal unit vector ( N )</td>
<td>Inches</td>
</tr>
<tr>
<td>HN(3)</td>
<td>( \bar{N} )</td>
<td>( x, y, ) and ( z ) components of the normal unit vector ( N )</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Variable used to count the intersects with the plane surfaces</td>
<td>ND</td>
</tr>
<tr>
<td>IA</td>
<td>---</td>
<td>Pointer to the coordinates of the semi-major axis unit vector</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>---</td>
<td>Pointer to the coordinates of the height vector</td>
<td>ND</td>
</tr>
<tr>
<td>IN</td>
<td>---</td>
<td>Pointer to the coordinates of the normal unit vector</td>
<td>ND</td>
</tr>
<tr>
<td>IRL</td>
<td>---</td>
<td>Pointer to the coordinates of the semi-major axis radius of the base ellipse</td>
<td>ND</td>
</tr>
</tbody>
</table>
**LIST OF SYMBOLS AND ABBREVIATIONS**

**SUBROUTINE TEC (Continued)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>IR2</td>
<td>---</td>
<td>Pointer to the coordinates of the semi-minor axis radius of the base ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>IR3</td>
<td>---</td>
<td>Pointer to the value of the ratio of the base ellipse to the top ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Pointer to the coordinates of the vertex</td>
<td>ND</td>
</tr>
<tr>
<td>L1</td>
<td>---</td>
<td>Temporary storage location for surface number of RIN for a planar surface</td>
<td>ND</td>
</tr>
<tr>
<td>LO</td>
<td>---</td>
<td>Temporary storage location for surface number of ROUT for a planar surface</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Pointer to the location of TEC data in the ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>R1</td>
<td>R1</td>
<td>Length of the semi-major axis of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>R2</td>
<td>R2</td>
<td>Length of the semi-minor axis of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>R2SQ</td>
<td>$(R2)^2$</td>
<td>Square of the length of the semi-minor axis of the base ellipse</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>R3</td>
<td>R3</td>
<td>Length of the semi-major axis of the top ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>R4</td>
<td>R4</td>
<td>Length of the semi-minor axis of the top ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>RR</td>
<td>RR</td>
<td>Ratio of the larger to the smaller ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>S</td>
<td>---</td>
<td>Temporary storage location for RIN and ROUT when verifying intersections</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>S1</td>
<td>---</td>
<td>Temporary storage location for the distance to the entry intersect for a quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>S2</td>
<td>---</td>
<td>Temporary storage location for the distance to the exit intersect for a quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>S1</td>
<td>---</td>
<td>Temporary storage location for the distance to the entry intersect for a plane surface</td>
<td>Inches</td>
</tr>
<tr>
<td>S0</td>
<td>---</td>
<td>Temporary storage location for the distance to the exit intersect for a plane surface</td>
<td>Inches</td>
</tr>
<tr>
<td>T</td>
<td>---</td>
<td>Temporary storage location used to interchange the values of two variables that represent the two intersections with the quadratic surface</td>
<td>Inches</td>
</tr>
<tr>
<td>T1</td>
<td>---</td>
<td>Value of the quadratic equation for the smaller root</td>
<td>Inches</td>
</tr>
<tr>
<td>T2</td>
<td>---</td>
<td>Value of the quadratic equation for the larger root</td>
<td>Inches</td>
</tr>
<tr>
<td>TA</td>
<td>((V-\bar{X}B) \cdot \bar{A} + \gamma(\bar{A} \cdot \bar{H}))</td>
<td>Quantity used in solving the coefficient of 2S, (\lambda), and in solving the constant term, (\mu), in the equation (rS^2+2\lambda S+\mu=0)</td>
<td>Inches</td>
</tr>
<tr>
<td>TA1</td>
<td>(\alpha WB \cdot \bar{A} - \bar{H} \cdot \bar{A})</td>
<td>Quantity used in solving for the coefficients of (S^2) and 2S in the equation (rS^2+2\lambda S+\mu=0)</td>
<td>Inches</td>
</tr>
<tr>
<td>TA2</td>
<td>((V-\bar{X}B) \cdot \bar{A} - \beta WB \cdot \bar{A})</td>
<td>Quantity used in solving for the constant term (\mu) and the coefficient (rS^2+2\lambda S+\mu=0)</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>TAU</td>
<td>((R_1/R_2)^2)</td>
<td>Square of the ratio of the length of the semi-major axis of the base ellipse to the length of the semi-minor axis of the base ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>TB</td>
<td>((V-XB) - B + \gamma(H-B))</td>
<td>Quantity used in solving the coefficient of (2S, \lambda,) and in solving the constant term, (\mu,) in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>Inches</td>
</tr>
<tr>
<td>TB1</td>
<td>(\omega WB - B - H)</td>
<td>Quantity used in solving for (\lambda) and (\mu,) in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>Inches</td>
</tr>
<tr>
<td>TB2</td>
<td>((V-XB) \cdot B - WB \cdot B)</td>
<td>Quantity used in solving for (\lambda) and (\mu,) in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>Inches</td>
</tr>
<tr>
<td>TR2SQ</td>
<td>((R_1/R_2)^2(R_2)^2)</td>
<td>Quantity used in solving for (\lambda) and (\mu,) in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>Inches^2</td>
</tr>
<tr>
<td>TR4R2</td>
<td>((R_1/R_2)^2(R_2-R_4)^2)</td>
<td>Quantity used in solving for (\tau,) in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>Inches^2</td>
</tr>
<tr>
<td>TRR4R2</td>
<td>((R_1/R_2)^2(R_2)^2 - (R_1/R_2)^2(R_2)/(R_4))</td>
<td>Quantity used in solving for (\lambda,) in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>Inches^2</td>
</tr>
<tr>
<td>UM</td>
<td>(\mu)</td>
<td>Value of the constant term in the equation (\tau S^2 + 2\lambda S + \mu = 0)</td>
<td>ND</td>
</tr>
<tr>
<td>VXB(3)</td>
<td>(\overline{V-XB})</td>
<td>x, y, and z components of the vector (V-XB)</td>
<td>Inches</td>
</tr>
<tr>
<td>VXBDA</td>
<td>((V-XB) \cdot A)</td>
<td>Dot product of the vector (V-XB) and the semi-major axis unit vector (A) of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>VXBDB</td>
<td>((V-XB) \cdot B)</td>
<td>Dot product of the vector (V-XB) and the semi-minor axis unit vector (B) of the base ellipse</td>
<td>Inches</td>
</tr>
<tr>
<td>VXBDN</td>
<td>((V-XB) \cdot \bar{N})</td>
<td>Dot product of the vector (V-XB) and the normal unit vector (\bar{N})</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>WDA</td>
<td>WB·A</td>
<td>Dot product of the direction cosines of the ray unit vector WB and the semi-major axis unit vector A of the base ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>WBD</td>
<td>WB·B</td>
<td>Dot product of the direction cosines of the ray unit vector WB and the semi-minor axis unit vector B of the base ellipse</td>
<td>ND</td>
</tr>
<tr>
<td>WDN</td>
<td>WB·N</td>
<td>Dot product of the direction cosines of the ray unit vector WB and the normal unit vector N</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALPHA</td>
<td>α</td>
<td>Value of α from matrix equation: $\alpha(U-W)+\beta(V-W)-S \cdot WB = XB-W$</td>
<td>ND*</td>
</tr>
<tr>
<td>BETA</td>
<td>β</td>
<td>Value of β from matrix equation: $\alpha(U-W)+\beta(V-W)-S \cdot WB = XB-W$</td>
<td>ND</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>Denominator of matrix equation: $\alpha(U-W)+\beta(V-W)-S \cdot WB = XB-W$</td>
<td>ND</td>
</tr>
<tr>
<td>DALPHA</td>
<td></td>
<td>Numerator for solving for α in matrix equation: $\alpha(U-W)+\beta(V-W)-S \cdot WB = XB-W$</td>
<td>ND</td>
</tr>
<tr>
<td>DBETA</td>
<td></td>
<td>Numerator for solving for β in matrix equation: $\alpha(U-W)+\beta(V-W)-S \cdot WB = XB-W$</td>
<td>ND</td>
</tr>
<tr>
<td>DS</td>
<td></td>
<td>Numerator for solving for S in matrix equation: $\alpha(U-W)+\beta(V-W)-S \cdot WB = XB-W$</td>
<td>ND</td>
</tr>
<tr>
<td>GAMMA</td>
<td>1-α-β</td>
<td>Value of $\gamma$ from equation: $XP=XB+S \cdot WB = \alpha \cdot U + \beta \cdot V + \gamma \cdot W$</td>
<td>ND</td>
</tr>
<tr>
<td>HIT(20)</td>
<td></td>
<td>Array for storing intersect distances (largest to smallest) while solving ray intersection with ARS</td>
<td>Inches</td>
</tr>
<tr>
<td>I</td>
<td></td>
<td>DO loop index for hit number</td>
<td>ND</td>
</tr>
<tr>
<td>ISURF(20)</td>
<td></td>
<td>Array for storing intersected triangle number of ARS. Positive or negative integer for exit or entry intersect, respectively</td>
<td>ND</td>
</tr>
<tr>
<td>IT</td>
<td></td>
<td>DO loop index for triangle number of ARS</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td></td>
<td>Number of intersects save area when intersect with larger distance has been found</td>
<td>ND</td>
</tr>
<tr>
<td>JSURF</td>
<td></td>
<td>Intersected triangle number</td>
<td>ND</td>
</tr>
</tbody>
</table>

*Non-dimensional*
# LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

## SUBROUTINE ARS (Continued)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>---</td>
<td>DO loop index for clearing RIN/ROUT ARS section of ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Pointer for data in ARS section of ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOCARS</td>
<td>---</td>
<td>Pointer to beginning location of ARS data in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOCHTS</td>
<td>---</td>
<td>Pointer to location of hit table in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>L1</td>
<td>---</td>
<td>Lower limit of DO loop for clearing hit table in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>L2</td>
<td>---</td>
<td>Upper limit of DO loop for clearing hit table in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>NHIT</td>
<td>---</td>
<td>Number of intersects of current ray with ARS</td>
<td>ND</td>
</tr>
<tr>
<td>NT</td>
<td>---</td>
<td>Total number of possible combinations of points that could form triangles on ARS surface</td>
<td>ND</td>
</tr>
<tr>
<td>ORMAL(3,20)</td>
<td>WN(3)</td>
<td>Intermediate storage array for storing normal coordinates to intersected triangle</td>
<td>ND</td>
</tr>
<tr>
<td>S</td>
<td>S</td>
<td>Distance from current origin of ray to intersected triangle</td>
<td>Inches</td>
</tr>
<tr>
<td>UW(3)</td>
<td>U-W</td>
<td>Array for storing coordinates of vector U-W for computed triangle</td>
<td>Inches</td>
</tr>
<tr>
<td>VW(3)</td>
<td>V-W</td>
<td>Array for storing coordinates of vector V-W for computed triangle</td>
<td>Inches</td>
</tr>
<tr>
<td>W(3)</td>
<td>W</td>
<td>Array for storing coordinates of vector W for computed triangle</td>
<td>Inches</td>
</tr>
<tr>
<td>WN(3)</td>
<td>---</td>
<td>Array for storing coordinates of normal vector to surface of computed triangle</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>---------</td>
</tr>
<tr>
<td>WXB(3)</td>
<td>W-XB</td>
<td>Array for storing coordinates of vector from point W of computed triangle to current origin of ray, XB</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
### (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>COEF(4)</td>
<td>B,C,D,E</td>
<td>Four-element array for coefficients of quadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index for RT array</td>
<td>ND</td>
</tr>
<tr>
<td>IN</td>
<td>---</td>
<td>Location of coordinates of normal of torus in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IR1</td>
<td>---</td>
<td>Location of major radius of torus in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IR2</td>
<td>---</td>
<td>Location of minor radius of torus in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Location of coordinates of center of torus in ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Location of packed word with radii of torus</td>
<td>ND</td>
</tr>
<tr>
<td>NR</td>
<td>---</td>
<td>Number of real intersects with torus</td>
<td>ND</td>
</tr>
<tr>
<td>R1</td>
<td>( r_1 )</td>
<td>Major radius of torus</td>
<td>Inches</td>
</tr>
<tr>
<td>R1SQ</td>
<td>( r_1^2 )</td>
<td>Square of major radius of torus</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>R2</td>
<td>( r_2 )</td>
<td>Minor radius of torus</td>
<td>Inches</td>
</tr>
<tr>
<td>R2SQ</td>
<td>( r_2^2 )</td>
<td>Square of minor radius of torus</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>RSAVE</td>
<td>---</td>
<td>Scalar quantity for shifting (XB-V) along ray to insure correct solution of quartic equation</td>
<td>ND</td>
</tr>
<tr>
<td>RT(4)</td>
<td>---</td>
<td>Four-element array for roots of quartic equation</td>
<td>Inches</td>
</tr>
<tr>
<td>T</td>
<td>---</td>
<td>Temporary storage location for exchanging values in two other storage locations</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

### SUBROUTINE TOR (Concluded)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>TERM</td>
<td>((\bar{X}B - \bar{C})^2 - r_1^2 - r_2^2)</td>
<td>Intermediate value for solving coefficients of quartic equation</td>
<td>Inches²</td>
</tr>
<tr>
<td>WDN</td>
<td>(\bar{W}B \cdot \bar{N})</td>
<td>Dot product of normal with direction cosines of ray</td>
<td>Inches</td>
</tr>
<tr>
<td>XBC(3)</td>
<td>((\bar{X}B - \bar{C}))</td>
<td>Coordinates of vector from center of torus to ray origin</td>
<td>Inches</td>
</tr>
<tr>
<td>XBCDN</td>
<td>((\bar{X}B - \bar{C}) \cdot \bar{N})</td>
<td>Dot product of vector from center of torus to ray origin with normal to torus</td>
<td>Inches</td>
</tr>
<tr>
<td>XBCDW</td>
<td>((\bar{X}B - \bar{C}) \cdot \bar{W}B)</td>
<td>Dot product of vector from center of torus to ray origin with direction cosines of ray</td>
<td>Inches</td>
</tr>
<tr>
<td>XBCXBC</td>
<td>((\bar{X}B - \bar{C})^2)</td>
<td>Dot product of vector from center of torus to ray origin with itself</td>
<td>Inches²</td>
</tr>
<tr>
<td>XN(3)</td>
<td>(\bar{N})</td>
<td>Coordinates of unit normal vector of torus</td>
<td>ND</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS

**(SIMULATION MODEL)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>$\sqrt{\frac{a^2}{4} - b + 2w}$</td>
<td>Square root of coefficient of $x^2$ of biquadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>ASQ</td>
<td>$\frac{a^2}{4} - b + 2w$</td>
<td>Coefficient of $x^2$ of biquadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>B</td>
<td>$\sqrt{w-d}$</td>
<td>Square root of constant term of biquadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>BSQ</td>
<td>$b^2$</td>
<td>Coefficient of constant term of biquadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>C(4)</td>
<td>$a, b, c, d$</td>
<td>Coefficients of quartic equation</td>
<td>ND</td>
</tr>
<tr>
<td>CİSQ</td>
<td>$a^2$</td>
<td>Square of coefficient of $x^3$ of quartic equation</td>
<td>ND</td>
</tr>
<tr>
<td>DISC</td>
<td>See Equations (203) and (204)</td>
<td>Discriminate of quadratic Equations (203) and (204)</td>
<td>Inches$^2$</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>DO loop index for number of roots from cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Number of real roots</td>
<td>ND</td>
</tr>
<tr>
<td>NN</td>
<td>---</td>
<td>Number of real roots in cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>R(4)</td>
<td>$x_{1,2,3,4}$</td>
<td>Array for storing roots of quartic equation</td>
<td>ND</td>
</tr>
<tr>
<td>REAL</td>
<td>$-\frac{a}{2} + e$</td>
<td>Real part of quadratic equation</td>
<td>Inches</td>
</tr>
<tr>
<td>ROOT</td>
<td>---</td>
<td>Real root of cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>RR(3)</td>
<td>---</td>
<td>Three-element array with roots of cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>SQROOT</td>
<td>$\sqrt{\text{DISC}}$</td>
<td>Radical part of quadratic equation</td>
<td>Inches</td>
</tr>
<tr>
<td>T</td>
<td>$\frac{a^2}{4} - b$</td>
<td>Part of ASQ</td>
<td>ND</td>
</tr>
<tr>
<td>TWOAB</td>
<td>$aw - c$</td>
<td>Coefficient of $x$ of biquadratic equation</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>A</td>
<td>$\sqrt[3]{\frac{q}{2} + \sqrt{Q}}$</td>
<td>Expression for solving the cubic equation for one real root and two conjugate complex roots</td>
<td>ND</td>
</tr>
<tr>
<td>AB</td>
<td>$A + B$</td>
<td>Expression for solving the cubic equation for one real root and two conjugate complex roots</td>
<td>ND</td>
</tr>
<tr>
<td>ACU</td>
<td>$-\frac{q}{2} + \sqrt{Q}$</td>
<td>Quantity under the radical of A</td>
<td>ND</td>
</tr>
<tr>
<td>B</td>
<td>$\sqrt[3]{\frac{q}{2} - \sqrt{Q}}$</td>
<td>Expression for solving the cubic equation for one real root and two conjugate complex roots</td>
<td>ND</td>
</tr>
<tr>
<td>BCU</td>
<td>$-\frac{q}{2} - \sqrt{Q}$</td>
<td>Quantity under the radical of B</td>
<td>ND</td>
</tr>
<tr>
<td>C</td>
<td>a, b, or c</td>
<td>Coefficients of $x^2$, x, and the constant term</td>
<td>ND</td>
</tr>
<tr>
<td>CLSQ</td>
<td>$a^2$</td>
<td>Square of the coefficient of $x^2$</td>
<td>ND</td>
</tr>
<tr>
<td>C3</td>
<td>$\frac{a}{3}$</td>
<td>Value of the constant term of the cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>DISC</td>
<td>$4p^3 + 27q^2$</td>
<td>Expression for solving for Q</td>
<td>ND</td>
</tr>
<tr>
<td>HALFQ</td>
<td>$q/2$</td>
<td>One-half the value of Q</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Integer variable that represents the number of real roots in the cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>P</td>
<td>$b - \frac{a^2}{3}$</td>
<td>Expression for solving for Q</td>
<td>ND</td>
</tr>
<tr>
<td>PHI3</td>
<td>$\phi/3$</td>
<td>One-third the angle</td>
<td>Radians</td>
</tr>
<tr>
<td>Q</td>
<td>$(\frac{p}{3})^3 + (\frac{a}{2})^2$</td>
<td>Expression used to solve for A and B</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>R(1)</td>
<td>$x_1$</td>
<td>Argument for passing real root cubic equation</td>
<td>ND</td>
</tr>
<tr>
<td>R(2)</td>
<td>$x_2$</td>
<td>Argument for passing real root, or real part of complex root</td>
<td>ND</td>
</tr>
<tr>
<td>R(3)</td>
<td>$x_3$</td>
<td>Argument for passing real root, or imaginary part of complex root</td>
<td>ND</td>
</tr>
<tr>
<td>SQROOT</td>
<td>$\sqrt{Q}$</td>
<td>Expression used to solve for A and B</td>
<td>ND</td>
</tr>
<tr>
<td>T</td>
<td>$\sqrt{-p/3}$</td>
<td>Expression for solving for three real roots with two equal roots</td>
<td>ND</td>
</tr>
<tr>
<td>TT</td>
<td>$2\sqrt{-p/3}$</td>
<td>Expression for solving for two equal roots when there are three real roots</td>
<td>ND</td>
</tr>
</tbody>
</table>
# LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I3</td>
<td>---</td>
<td>Packed word from the Lth word of the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>J1</td>
<td>---</td>
<td>Integer data item in the 15 bits previous to the last 15 bits of the two-item packed word</td>
<td>ND</td>
</tr>
<tr>
<td>J2</td>
<td>---</td>
<td>Integer data item in the last 15 bits of the two-item packed word</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Pointer to the location in the MASTER array of the packed word</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>I2</td>
<td>---</td>
<td>Packed word containing only J1 and J2 after J3 has been shifted out</td>
<td>ND</td>
</tr>
<tr>
<td>I3</td>
<td>---</td>
<td>Packed word from the Lth word of the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>J1</td>
<td>---</td>
<td>Integer data item in the six bits previous to the six bits of J2</td>
<td>ND</td>
</tr>
<tr>
<td>J2</td>
<td>---</td>
<td>Integer data item in the six bits previous to the last 15 bits of J3</td>
<td>ND</td>
</tr>
<tr>
<td>J3</td>
<td>---</td>
<td>Integer data item in the last 15 bits of the packed word</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Pointer to the location in the MASTER array of the packed word</td>
<td>ND</td>
</tr>
</tbody>
</table>
## List of Symbols and Abbreviations

### Simulation Model

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I2</td>
<td>---</td>
<td>Packed word containing only J1 and J2 after J3 has been shifted out</td>
<td>ND</td>
</tr>
<tr>
<td>I3</td>
<td>---</td>
<td>Packed word from the Lth location of the ITR array</td>
<td>ND</td>
</tr>
<tr>
<td>J1</td>
<td>---</td>
<td>12 bits of the left integer data item of the packed word</td>
<td>ND</td>
</tr>
<tr>
<td>J2</td>
<td>---</td>
<td>Middle 12 bits of the integer data item of the packed word</td>
<td>ND</td>
</tr>
<tr>
<td>J3</td>
<td>---</td>
<td>Integer data item in the last 12 bits of the packed word</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Pointer to the location in the ITR array of the packed word</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>RAN</td>
<td>---</td>
<td>Random number between zero and one returned by Function URAN31</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1</td>
<td>---</td>
<td>Temporary fixed point storage location for computing the random number</td>
<td>ND</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Integer variable that represents the argument of the function</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Temporary integer storage location used for computing the random number</td>
<td>ND</td>
</tr>
<tr>
<td>URAN31</td>
<td>---</td>
<td>Value of the random number between zero and one</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>ANSWER(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the resultant vector from the cross product of two other vectors</td>
<td>Inches</td>
</tr>
<tr>
<td>FIRST(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the first argument vector</td>
<td>Inches</td>
</tr>
<tr>
<td>SECOND(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the second argument vector</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>DOT</td>
<td>---</td>
<td>Resultant scalar quantity from the dot product of two vectors</td>
<td>Inches²</td>
</tr>
<tr>
<td>FIRST(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the first argument vector</td>
<td>Inches</td>
</tr>
<tr>
<td>SECOND(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the second argument vector</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>TEMP</td>
<td>---</td>
<td>Scalar length of the vector</td>
<td>Inches</td>
</tr>
<tr>
<td>V(3)</td>
<td>---</td>
<td>Coordinates of either the original vector or the resultant unit vector</td>
<td>Inches</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

#### SUBROUTINE XDIST

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>XDIST</td>
<td>---</td>
<td>Distance between the two given points</td>
<td>Inches</td>
</tr>
<tr>
<td>XSUM</td>
<td>---</td>
<td>Temporary storage area for summing the coordinate distances between the two points</td>
<td>Inches²</td>
</tr>
<tr>
<td>XA(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the first point</td>
<td>Inches</td>
</tr>
<tr>
<td>XB(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the second point</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>DIS</td>
<td>---</td>
<td>Scalar distance between points XA and XB</td>
<td>Inches</td>
</tr>
<tr>
<td>WA(3)</td>
<td>---</td>
<td>Three-element array for the direction cosines or unit vector coordinates of the vector from point XA to XB</td>
<td>Inches</td>
</tr>
<tr>
<td>XA(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the first point</td>
<td>Inches</td>
</tr>
<tr>
<td>XB(3)</td>
<td>---</td>
<td>Three-element array for the coordinates of the second point</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

### SUBROUTINE TROPIC

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSPHI</td>
<td></td>
<td>Cosine of a random angle $\phi$</td>
<td>ND</td>
</tr>
<tr>
<td>CSTHT</td>
<td></td>
<td>Cosine of a random angle $\theta$</td>
<td>ND</td>
</tr>
<tr>
<td>SNPHI</td>
<td></td>
<td>Sine of a random angle $\phi$</td>
<td>ND</td>
</tr>
<tr>
<td>SNTHT</td>
<td></td>
<td>Sine of a random angle $\theta$</td>
<td>ND</td>
</tr>
<tr>
<td>T</td>
<td></td>
<td>Sum of the squares of two random numbers that is less than or equal to 1.0</td>
<td>ND</td>
</tr>
<tr>
<td>WP(3)</td>
<td></td>
<td>Three-element array containing the direction cosines of a random angle</td>
<td>ND</td>
</tr>
<tr>
<td>X1</td>
<td></td>
<td>Random number between zero and one</td>
<td>ND</td>
</tr>
<tr>
<td>X2</td>
<td></td>
<td>Random number between zero and one</td>
<td>ND</td>
</tr>
<tr>
<td>X1S</td>
<td></td>
<td>Square of random number X1</td>
<td>ND</td>
</tr>
<tr>
<td>X2S</td>
<td></td>
<td>Square of random number X2</td>
<td>ND</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS

**SUBROUTINE S**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>---</td>
<td>Number of the RPP</td>
<td>ND</td>
</tr>
<tr>
<td>L</td>
<td>---</td>
<td>Location of the pointer data for a given side of a given RPP</td>
<td>ND</td>
</tr>
<tr>
<td>LL</td>
<td>---</td>
<td>Location of the coordinate for a given side of a given RPP</td>
<td>ND</td>
</tr>
<tr>
<td>N</td>
<td>---</td>
<td>Surface number of the RPP</td>
<td>ND</td>
</tr>
<tr>
<td>S</td>
<td>---</td>
<td>Coordinate of the given side of the given RPP number</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS
(SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>---</td>
<td>Index for a DO loop to test each potential abutting RPP</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>---</td>
<td>Number of a potential abutting RPP unpacked from the left position of the two-item packed word in the abutting RPP section</td>
<td>ND</td>
</tr>
<tr>
<td>I2</td>
<td>---</td>
<td>Number of a potential abutting RPP unpacked from the left position of the two-item packed word in the abutting RPP section</td>
<td>ND</td>
</tr>
<tr>
<td>IRP</td>
<td>---</td>
<td>Number of the potential RPP, or the RPP number returned to the calling program</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Index to represent an x, y, or z coordinate in a DO loop</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Variable used to represent the location of abutting RPP's in the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LOCAT</td>
<td>---</td>
<td>Pointer to the location of the abutting RPP list in the MASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>LS</td>
<td>---</td>
<td>Control variable used to prevent a specific intersect coordinate from being tested with its respective plane coordinate</td>
<td>ND</td>
</tr>
<tr>
<td>LSURF</td>
<td>---</td>
<td>Surface number of the RPP where the intersect occurs</td>
<td>ND</td>
</tr>
<tr>
<td>M</td>
<td>---</td>
<td>Control integer used to determine which of two packed abutting RPP's is to be tested</td>
<td>ND</td>
</tr>
<tr>
<td>NC</td>
<td>---</td>
<td>Total number of abutting RPP's</td>
<td>ND</td>
</tr>
<tr>
<td>XP</td>
<td>---</td>
<td>Coordinates of the intersect</td>
<td>Inches</td>
</tr>
</tbody>
</table>
## LIST OF SYMBOLS AND ABBREVIATIONS
### (SIMULATION MODEL)

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOD</td>
<td>---</td>
<td>Horizontal dimension of cell on front plane of box</td>
<td>Inches</td>
</tr>
<tr>
<td>DSP</td>
<td>---</td>
<td>Vector from first ray origin to second ray origin in given column</td>
<td>Inches</td>
</tr>
<tr>
<td>DT</td>
<td>---</td>
<td>Vertical dimension of cell on front plane of box</td>
<td>Inches</td>
</tr>
<tr>
<td>IR1</td>
<td>---</td>
<td>Region number of pre-computed volume</td>
<td>ND</td>
</tr>
<tr>
<td>IRJ</td>
<td>---</td>
<td>Region number save area</td>
<td>ND</td>
</tr>
<tr>
<td>IRPRIM</td>
<td>---</td>
<td>Region number returned by Sub-routine Gl</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Index for column or plane number</td>
<td>ND</td>
</tr>
<tr>
<td>J1</td>
<td>---</td>
<td>Index for x, y, or z coordinate</td>
<td>ND</td>
</tr>
<tr>
<td>JIR</td>
<td>---</td>
<td>Region number save area</td>
<td>ND</td>
</tr>
<tr>
<td>Jl</td>
<td>---</td>
<td>Number of vertical cells</td>
<td>ND</td>
</tr>
<tr>
<td>N2</td>
<td>---</td>
<td>Number of horizontal cells</td>
<td>ND</td>
</tr>
<tr>
<td>S1</td>
<td>---</td>
<td>Distance to next region returned by Subroutine Gl</td>
<td>Inches</td>
</tr>
<tr>
<td>SUMV</td>
<td>---</td>
<td>Sum of the computed volumes in box</td>
<td>Inches³</td>
</tr>
<tr>
<td>TESTDN</td>
<td>---</td>
<td>Vertical distance to next region</td>
<td>Inches</td>
</tr>
<tr>
<td>TESTOV</td>
<td>---</td>
<td>Horizontal distance to next region</td>
<td>Inches</td>
</tr>
<tr>
<td>VASTER</td>
<td>---</td>
<td>Array for accumulating ray distance through each region within box</td>
<td>Inches</td>
</tr>
<tr>
<td>VR</td>
<td>---</td>
<td>Pre-computed volume of given region</td>
<td>Inches³</td>
</tr>
<tr>
<td>WAB</td>
<td>---</td>
<td>Direction cosines of vector from plane to back plane of box</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>WOB</td>
<td>---</td>
<td>Direction cosines of horizontal vector from vertex across front plane of box</td>
<td>ND</td>
</tr>
<tr>
<td>WTB</td>
<td>---</td>
<td>Direction cosines of vertical vector from vertex across front plane of box</td>
<td>ND</td>
</tr>
<tr>
<td>XA(3)</td>
<td>---</td>
<td>Coordinates of point on back plane</td>
<td>Inches</td>
</tr>
<tr>
<td>XO(3)</td>
<td>---</td>
<td>Coordinates of lower left corner of front plane of box</td>
<td>Inches</td>
</tr>
<tr>
<td>XP(3)</td>
<td>---</td>
<td>Coordinates of intersect with next region</td>
<td>Inches</td>
</tr>
<tr>
<td>XPARC</td>
<td>---</td>
<td>Percent error of computed volume and pre-computed volume of given region</td>
<td>ND</td>
</tr>
<tr>
<td>XT(3)</td>
<td>---</td>
<td>Coordinates of upper right corner of front plane of box</td>
<td>Inches</td>
</tr>
<tr>
<td>XTEMP(3)</td>
<td>---</td>
<td>Temporary storage for direction cosines</td>
<td>ND</td>
</tr>
<tr>
<td>XV(3)</td>
<td>---</td>
<td>Coordinates of vertex of box</td>
<td>Inches</td>
</tr>
<tr>
<td>XVDIS</td>
<td>---</td>
<td>Distance from front plane to back plane</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>Azimuth angle of grid plane</td>
<td>Degrees</td>
</tr>
<tr>
<td>AR</td>
<td>a</td>
<td>Azimuth angle of grid plane</td>
<td>Radians</td>
</tr>
<tr>
<td>AREAC</td>
<td></td>
<td>Area of cell in grid plane</td>
<td>Inches², Feet², Centimeters², or Meters²</td>
</tr>
<tr>
<td>AREAUN</td>
<td></td>
<td>Measurement units of presented area</td>
<td>ND</td>
</tr>
<tr>
<td>BLANK</td>
<td></td>
<td>Two Hollerith blanks</td>
<td>ND</td>
</tr>
<tr>
<td>CA</td>
<td>cos a</td>
<td>Cosine of azimuth angle</td>
<td>ND</td>
</tr>
<tr>
<td>CE</td>
<td>cos θ</td>
<td>Cosine of elevation angle</td>
<td>ND</td>
</tr>
<tr>
<td>CELL2</td>
<td>D/2</td>
<td>Half the dimension of a cell side</td>
<td>Inches, Feet, Centimeters, or Meters</td>
</tr>
<tr>
<td>CELLUN</td>
<td></td>
<td>Measurement units of cell</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(1,1)</td>
<td></td>
<td>Conversion factor for converting square inches to square inches (= 1)</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT (1,2)</td>
<td></td>
<td>Conversion factor for converting square inches to square feet (≈ 0.006944444444444)</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(1,3)</td>
<td></td>
<td>Conversion factor for converting square inches to square centimeters (≈ 6.451625806)</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(1,4)</td>
<td></td>
<td>Conversion factor for converting square inches to square meters (≈ 0.0006451625806)</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>CONVRT(2,1)</td>
<td>---</td>
<td>Conversion factor for converting square feet to square inches ((= 144))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(2,2)</td>
<td>---</td>
<td>Conversion factor for converting square feet to square feet ((= 1))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(2,3)</td>
<td>---</td>
<td>Conversion factor for converting square feet to square centimeters ((= 929.0341161))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(2,4)</td>
<td>---</td>
<td>Conversion factor for converting square feet to square meters ((= 0.09290341161))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(3,1)</td>
<td>---</td>
<td>Conversion factor for converting square centimeters to square inches ((= 0.15499969))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(3,2)</td>
<td>---</td>
<td>Conversion factor for converting square centimeters to square feet ((= 0.001076386736))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(3,3)</td>
<td>---</td>
<td>Conversion factor for converting square centimeters to square centimeters ((= 1))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(3,4)</td>
<td>---</td>
<td>Conversion factor for converting square centimeters to square meters ((= 0.00001))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(4,1)</td>
<td>---</td>
<td>Conversion factor for converting square meters to square inches ((= 1549.9969))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(4,2)</td>
<td>---</td>
<td>Conversion factor for converting square meters to square feet ((= 10.7636736))</td>
<td>ND</td>
</tr>
<tr>
<td>CONVRT(4,3)</td>
<td>---</td>
<td>Conversion factor for converting square meters to square centimeters ((= 10000))</td>
<td>ND</td>
</tr>
</tbody>
</table>
### LIST OF SYMBOLS AND ABBREVIATIONS

**SUBROUTINE AREA (Continued)**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONVRT(4,4)</td>
<td>---</td>
<td>Conversion factor for converting square meters to square meters (= 1)</td>
<td>ND</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
<td>Elevation angle of grid plane</td>
<td>Degrees</td>
</tr>
<tr>
<td>ENGTH</td>
<td>ENGTH</td>
<td>Back-off distances or origin from grid plane</td>
<td>Inches</td>
</tr>
<tr>
<td>ER</td>
<td>θ</td>
<td>Elevation angle of grid plane</td>
<td>Radians</td>
</tr>
<tr>
<td>GROUND</td>
<td>---</td>
<td>z coordinate of ground level</td>
<td>Inches</td>
</tr>
<tr>
<td>HHBB</td>
<td>---</td>
<td>Two Hollerith blanks for testing for blank fields on card</td>
<td>ND</td>
</tr>
<tr>
<td>HHCM</td>
<td>---</td>
<td>Hollerith code for centimeters (CM)</td>
<td>ND</td>
</tr>
<tr>
<td>HHFT</td>
<td>---</td>
<td>Hollerith code for feet (FT)</td>
<td>ND</td>
</tr>
<tr>
<td>HHIN</td>
<td>---</td>
<td>Hollerith code for inches (IN)</td>
<td>ND</td>
</tr>
<tr>
<td>HHMB</td>
<td>---</td>
<td>Hollerith code for meters (M)</td>
<td>ND</td>
</tr>
<tr>
<td>I</td>
<td>---</td>
<td>Index for representing component code</td>
<td>ND</td>
</tr>
<tr>
<td>ICODE</td>
<td>---</td>
<td>Component code of region material</td>
<td>ND</td>
</tr>
<tr>
<td>IDENT</td>
<td>---</td>
<td>Identification code of region material</td>
<td>ND</td>
</tr>
<tr>
<td>IH</td>
<td>---</td>
<td>Random number for computing random horizontal point in given cell</td>
<td>ND</td>
</tr>
<tr>
<td>II</td>
<td>---</td>
<td>Variable for representing row number of grid cell</td>
<td>ND</td>
</tr>
<tr>
<td>IRPRIM</td>
<td>---</td>
<td>Next region number returned by Subroutine GI</td>
<td>ND</td>
</tr>
<tr>
<td>IRSTRT</td>
<td>---</td>
<td>Starting region of grid rays</td>
<td>ND</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IV</td>
<td>---</td>
<td>Random number for computing random vertical point in given cell</td>
<td>ND</td>
</tr>
<tr>
<td>J</td>
<td>---</td>
<td>Variable for representing column number of grid cell</td>
<td>ND</td>
</tr>
<tr>
<td>KK</td>
<td>---</td>
<td>Index for representing cell number of grid</td>
<td>ND</td>
</tr>
<tr>
<td>KL</td>
<td>---</td>
<td>Total number of cells in grid</td>
<td>ND</td>
</tr>
<tr>
<td>LAREA</td>
<td>---</td>
<td>Beginning location of presented areas indexed by component code</td>
<td>ND</td>
</tr>
<tr>
<td>LAREAL</td>
<td>---</td>
<td>Last location of presented areas indexed by component code</td>
<td>ND</td>
</tr>
<tr>
<td>LOC</td>
<td>---</td>
<td>Location of specific data in MASTER-ASTER array</td>
<td>ND</td>
</tr>
<tr>
<td>NHIT</td>
<td>---</td>
<td>Number of rays that hit target</td>
<td>ND</td>
</tr>
<tr>
<td>NSTART</td>
<td>---</td>
<td>Beginning grid cell number</td>
<td>ND</td>
</tr>
<tr>
<td>NX</td>
<td>$N_x$</td>
<td>Number of horizontal cells in grid plane</td>
<td>ND</td>
</tr>
<tr>
<td>NY</td>
<td>$N_y$</td>
<td>Number of vertical cells in grid plane</td>
<td>ND</td>
</tr>
<tr>
<td>RADIAN</td>
<td>$R$</td>
<td>One degree expressed in radians</td>
<td>Radians</td>
</tr>
<tr>
<td>SA</td>
<td>$\sin \alpha$</td>
<td>Sine of azimuth angle</td>
<td>ND</td>
</tr>
<tr>
<td>SE</td>
<td>$\sin \theta$</td>
<td>Sine of elevation angle</td>
<td>ND</td>
</tr>
<tr>
<td>SUMA</td>
<td>---</td>
<td>Total presented area of target</td>
<td>Inches$^2$, Feet$^2$, Centimeters$^2$, or Meters$^2$</td>
</tr>
<tr>
<td>TYPEUN(4)</td>
<td>---</td>
<td>Four-element array containing Hollerith codes IN, FT, CM, and M, respectively</td>
<td>ND</td>
</tr>
</tbody>
</table>
## Subroutine AREA (Concluded)  
**Simulation Model**

<table>
<thead>
<tr>
<th>Symbol or Abbreviation</th>
<th>Equivalent in Math Model</th>
<th>Definition</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>WB(3)</td>
<td>$\bar{W}B$</td>
<td>Direction cosines of ray</td>
<td>ND</td>
</tr>
<tr>
<td>WP(3)</td>
<td>---</td>
<td>Isotropic random direction cosines returned by Subroutine TROPIC</td>
<td>ND</td>
</tr>
<tr>
<td>XB(3)</td>
<td>$\bar{X}B$</td>
<td>Coordinates of origin of ray</td>
<td>Inches</td>
</tr>
<tr>
<td>XBS(3)</td>
<td>$\bar{X}_p$</td>
<td>Coordinates of original position of origin on grid cell before back-off</td>
<td>Inches</td>
</tr>
<tr>
<td>XSHIFT</td>
<td>$XSHIFT$</td>
<td>Distance target origin and grid plane center is effectively shifted in the X direction</td>
<td>Inches</td>
</tr>
<tr>
<td>YSHIFT</td>
<td>$YSHIFT$</td>
<td>Distance target origin and grid plane center is effectively shifted in the Y direction</td>
<td>Inches</td>
</tr>
<tr>
<td>ZSHIFT</td>
<td>$ZSHIFT$</td>
<td>Distance target origin and grid plane center is effectively shifted in the Z direction</td>
<td>Inches</td>
</tr>
<tr>
<td>Symbol or Abbreviation</td>
<td>Equivalent in Math Model</td>
<td>Definition</td>
<td>Units</td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------</td>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>IRAY</td>
<td>---</td>
<td>Index for number of rays to be processed</td>
<td>ND</td>
</tr>
<tr>
<td>IRFIN</td>
<td>---</td>
<td>Region number of end point</td>
<td>ND</td>
</tr>
<tr>
<td>IRSTRT</td>
<td>---</td>
<td>Region number of starting point</td>
<td>ND</td>
</tr>
<tr>
<td>NRAYS</td>
<td>---</td>
<td>Total number of rays to be processed</td>
<td>ND</td>
</tr>
<tr>
<td>RANGE</td>
<td>---</td>
<td>Distance between two points</td>
<td>Inches</td>
</tr>
<tr>
<td>SI</td>
<td>---</td>
<td>Distance to next region</td>
<td>Inches</td>
</tr>
<tr>
<td>XBF</td>
<td>---</td>
<td>Coordinates of end point</td>
<td>Inches</td>
</tr>
<tr>
<td>XP</td>
<td>---</td>
<td>Coordinates of intersect with next region</td>
<td>Inches</td>
</tr>
</tbody>
</table>
SECTION IV

SOURCE LISTING

SIMULATION SOURCE DECK

This section contains a listing of the FORTRAN statements that make up the program deck (Figures 74 through 114).

SAMPLE PROBLEM DECK

Figure 115 shows a listing of the sample problem deck data. A description of the sample problem is contained in Volume I, User Manual.
DIMENSION A(6)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/GEOM/BASE,LRIN ROUT,LRIR,LRPL,PLF,IERR,DISS
COMMON/NCGEOM/NRPP,NTRIP,NSCAL,NCBODY,NRMAX,NTRIP,LSCAL,LREGO,
1 LOAT,LRIN,LRTO,LOD,LOCOX,150,130,LBODY,NC1N,VOLUME,ITAPE,NO,YES
COMMON/ENGEM/LEGEM
COMMON/TMP,XSC,XT,XC,TC,NC,XNC,NC
COMMON/TEST/A,T,E,A
COMMON/WRITE,WRITE
COMMON/TEST/WRIT
COMMON/INF,NO,YES
COMMON/IRAND,IRAND

EQUIVALENCE (ASTER, MASTER)

901 FORMAT(1H1,32H THIS IS THE 11 APR 69 VERSION OF /)
1 1H32M THE BRLSEC MAGIC PROGRAM
2 16H BEGIN EXECUTION
902 FORMAT(8I1)
903 FORMAT(8I1)
904 FORMAT(1H0,10X,42H THE TAPE 4 USED FOR THIS RUN HAS THE TITLE /)
1 10A6/
905 FORMAT(1H0,10X,16H ENTER GENI)
906 FORMAT(1H0,10X,16H LEAVING GENI)
907 FORMAT(1H0,35H TERMINATION ON GEOMETRY INPUT ERROR, 5X,N,iERR=15)
908 FORMAT(1H1,16H TESTG IS CALLED)
909 FORMAT(1H0,13H LEAVING TESTG)
910 FORMAT(1H0,24H REGION TYPE DATA FOLLOWS, 8X,6H,LIRFO=110/
1 1M,6H REGION,6X,4H,CODE,6X,4H,TYPE,6X,11H,DESCRIPTION/)
911 FORMAT(11I0,10X,6A6)
912 FORMAT(16I0,10X,9X,6A6)
913 FORMAT(1H0,23H NO ROOM FOR IDENT TABLE,5X,7H,LEGEM,=17,5X,
1 6H,LIRFO,=17)
914 FORMAT(1H0,32H WRITE TAPE 1 OPTION IS SPECIFIED)
915 FORMAT(1S5,10X,10A6)
916 FORMAT(1H1,16H ENTER VOLUM)
917 FORMAT(1H0,13H LEAVING VOLUM)
918 FORMAT(1H0,6H,999,9)
919 FORMAT(1H1,11H END OF CASE=15)
925 FORMAT(1H1,32H NUM OF ASPECT ANGLES FOR GRID IS=15)
927 FORMAT(10I1)
928 FORMAT(1H1,32H NUM OF ASPECT ANGLES FOR AREA IS=15)
929 FORMAT(1H0,31H NUMBER OF G1 ERRORS ENCOUNTERED=15)
930 FORMAT(1H0,31H NUMBER OF 0 ITEMS ENCOUNTERED=15)
999 FORMAT(1H0,10H END OF RUN)

C
IRAND=0
WRITE (6901)
WRITE (6902)

C
C
C
INITIALIZE CONSTANTS
C
IIS=2=15
IJO=2=15
PINF=1.0E50
NO=0
YES=1
IERR=0
ENTER AND INITIALIZE OPTION PARAMETERS

READ (5,903) IRDTP4, IWRTP4, ITESTG, IRAYSK, ICARD, IENTLYV, IVOLUM
IF (IRDTP4.NE.0) IRDTP4=IYES
IF (IWRTP4.NE.0) IWRTP4=IYES
IF (IESTG.NE.0) IESTG=IYES
IF (IRAYSK.NE.0) IRAYSK=IYES
IF (ICARD.NE.0) ICARD=IYES
IF (IENTLYV.NE.0) IENTLYV=IYES
IF (IVOLUM.NE.0) IVOLUM=IYES

ENTER TARGET GEOMETRY FROM INPUT TAPE 4

IF (IRDTP4.EQ.0) GOTO 10
READ (4) LBASE, LEGeom, NDOQ, (ASTEH(L), L=1, NDOQ), LBODY, LREGD, LRIN
WRITE (6,904) (IT(I), I=1, 10)
GOTO 20

CLEAR MASTER-ASTER ARRAY

10 DO 11 J=LBASE, NDOQ
ASTERC(J)=0.
11 CONTINUE

ENTER AND PROCESS TARGET GEOMETRY VIA SUBROUTINE GENI

WRITE (6,905)
CALL GENI
WRITE (6,906)
IF (IERR.LE.0) GOTO 12
WRITE (6,907) IERR
STOP

WRITE OUT TARGET GEOMETRY TO OUTPUT TAPE 4

12 IF (IWRTP4.EQ.0) GOTO 20
WRITE (6,908) LBASE, LEGeom, NDOQ, (ASTEH(L), L=1, NDOQ), LBODY, LREGD, LRIN
WRITE (6,910) LROT, LNO, LIRFO, NRPP, NBODY, NMAX, PINF, IT

CALL SUBROUTINE TESTG

20 IF (IESTG.EQ.0) GOTO 30
WRITE (6,909) ITESTG=NO

CALL SUBROUTINE VOLUM

30 IF (IVOLUM.EQ.0) GOTO 40
WRITE (6,916) IVOLUM=NO

FIG. 74. (Contd.)
WRITE (6,917)
IVOLUME=NO

REGION IDENTIFICATION DATA
IRC = REGION NUMBER
ICODE = ITEM CODE
IDENT = SPACE CODE AND SPECIAL IDENTIFICATION
SPECIAL IDENTIFICATION = 10*20*30*40*50*60*70*80*90
NO IDENT CODE=0 SKIRT=10 ARMOR=20 TARGET=30
SPACE CODES EXTERIOR VOLUME = 1
INTERIOR VOLUME = -1,1-9,...,91-99

LIRFO=NDQ=NRMAX=10
IF(LIRFO.GT.LEGEOM)GOTO 41
WRITE(6,913)LEGEOM,LIRFO
STOP
41 WRITE (6,910)LIRFO

ENTER AND STORE REGION ID DATA

READ (5,911)IRC,ICODE,IDENT,(A(I)),I=1,6)
IF(IRC.LE.0)GOTO 50
WRITE (6,912)IRC,ICODE,IDENT,(A(I)),I=1,6)
IDENT=IDENT+1
K=LIRFO-IRC
MASTER(K)=ICODE=I5*IDENT
GOTO 42

NOAA = NUMBER OF ASPECT ANGLES FOR SUBROUTINE GRID

ITAPEB = SUPPRESS PRINTING OPTION

IWOT = WRITE OPTION FOR TAPE

NAREA = NUMBER OF ASPECT ANGLES FOR SUBROUTINE AREA

READ (5,927)NOAA,IWOT,ITAPEB,NAREA
IF(IWOT.NE.0)IWOT=IYES
IF(ITAPEB.EQ.0)GOTO 51
ITAPEB=NO
GOTO 52
51 ITAPEB=IYES
52 IF(IWOT.EQ.NO)GOTO 60
REWIND 1
WRITE (6,914)
WRITE (6,915)NOAA+(I(I)+I=1,10)
60 IF(NOAA.LE.0)GOTO 70.
WRITE (6,925)NOAA

CALL SUBROUTINE GRID FOR EACH ASPECT ANGLE

DO 61 I=1,NOAA
IERR=0
IERRO=0
CALL GRID
IF(IWOT.EQ.IYES)WRITE (6,918)
WRITE (6,919)I
WRITE (6,929)IERR
WRITE (6,930)IERRO
61 CONTINUE

FIG. 74. (Contd.)
70 IF (NAREA .LE. 0) GOTO 99
       WRITE (*,920) NAREA
C   C
C   99 WRITE (*,999)
       STOP
C   C

C  CALL SUBROUTINE AREA FOR EACH ASPECT ANGLE
C   DO 71 I=1,NAREA
   IERR=0
   CALL AREA
   WRITE (*,910) I
    71 CONTINUE
C
FIG. 74. (Concluded)
SUBROUTINE GENI
DIMENSION ITY(11), IAN(R), IAA(8), FX(26), NBO(11)
1 NO(3), NO(3), N02(3), 04(3), TT(3), TT1(3), TT2(3)
DIMENSION MASTER(10000)
COMMON/ASTER(10000)
COMMON/GEOM/LBASE, RBASE, RIN, ROUT, LR1, LR2, RINP+ERR, DIST
COMMON/UNCENTER/NRP, NTRIP, NSCAL, NBO(7), NM, NLR, LSCAL, NREO,
1 LO, LIRN, LROT, IIO, LOC, I15, I30, LBOD, NASC, KLR, KLOOP
COMMON/CONT/ITEST, IEN, IER, ILEMT, IV embraces, IOT, ITAPE, NO, IIYES
COMMON/X, NQ
COMMON/UNCLE/NN, IC(4)
COMMON/RRPP/LRP, LDP, LUNIT
COMMON/ENGEM/LFGEOM

EQUIVALENCE (ASTER, MASTER)

C
901 FORMAT(1MO+2A$HSTART READING SOLID DATA)
902 FORMAT(1MO+10A$)
903 FORMAT(1MO+10A$)
904 FORMAT(7110)
905 FORMAT(4X+34MNO, OF RECTANGULAR PARALLELEPIPEDS II0/
1 4X+34MNO, OF SOLIDS +10/
2 4X+34MNO, NO. OF REGIONS +10/
906 FORMAT(1MO+5X+32MRECTANGULAR PARALLELEPIPED INPUT)
907 FORMAT(1MO+5X+32M DESCRIPTION OF SOLIDS)
912 FORMAT(3AI+A6+A6+10$)
913 FORMAT(1MO+64A$ + A3+A7H DOES NOT MATCH WITH AN IITY)
914 FORMAT(1MO+1X+3AI+A3+A6+3X+815)
915 FORMAT(1MO+64A$+1AI+2X+A3+A6+4X+6F12+5)
916 FORMAT(25X+6F12+5)
917 FORMAT(1MO+38HND MORE ROOM FOR SOLID DATA
1 5X+5MLBOT+I10+5X+4HNDQ+I10)
918 FORMAT(1MO+28HND MORE ROOM FOR SOLID DATA)
919 FORMAT(1MO+25HSLREGO+7M LREG=7M LENL=7M LRIN=7M LROT=7M
1 7M LIO=7M LGEOM=15+617)
920 FORMAT(1MO+36+23HREGION COMBINATION DATA)
921 FORMAT(I5+1X+9,122,15)
922 FORMAT(1MO+3MERROR IN DESCRIPTION OF REGION#5)
19H IN FIELD=12+5X, 2/NBOD NUM.GT.NRPP + NBOY)
923 FORMAT(I,5X+9( (H+12+15+11)+1/))
924 FORMAT(I=18+2X+9( (H+12+15+1)+1/))
925 FORMAT(1MO+3MILLEGAL OPERATOR IN ABOVE CARD+5X+12+5X)
1 9H IN FIELD=12)
926 FORMAT(1MO+29HERROR IN REGION INPUT IR=15,14H OR N.GT.NRMAX)
927 FORMAT(1MO+39M MORE ROOM FOR REGION DATA LOATA=I10,
1 5X+4HNDQ+I10)
928 FORMAT(1MO+26MFINISH READING REGION DATA)
929 FORMAT(14E9, 10H IS PART OF REGION#110)
930 FORMAT(24H FINISH CHECKING REGION #16)
931 FORMAT(1MO+36HND MORE ROOM FOR ENTER LEAVE TABLE=5X+)
1 6MLD=I10+5X+4HNDQ+I10+5X+6HNPASS+12+5X+3HREO, I10)
932 FORMAT(1MO+28H TOTAL ROOM FOR GEOMETRY DATA=5X+7MLGEOM=16)
933 FORMAT(1MO+5MENTER=1816/123X+1516)
934 FORMAT(1H+5MLEAVE=1816/123X+1516)
935 FORMAT(1MO+50X+1HREGION ARRAY OUTPUT/)
936 FORMAT(316+1X+E11+43H $)

FIG. 75. Source Listing, Subroutine GENI
937 FORMAT(/)
938 FORMAT(1H0=3X)FINISH A PASS OF ENTER LEAVE TABLE=15)
940 FORMAT(10X*6F10.5)
941 FORMAT(1H0=5X)TERMINATION ON BAD REGION DESCRIPTION
942 FORMAT(1H0=32X)ERROR IN DESCRIPTION OF BODY NUM=16/
  1 7M VECTOR=3F12.5*24M IS NOT PERPENDICULAR TO /
  2 7M VECTOR=3F12.5*
943 FORMAT(1H0=27X)ERROR IN DESCRIPTION OF TOR=5X*8MR2,6G1/1
944 FORMAT(1H0=27X)ERROR IN DESCRIPTION OF TRC=5X*7HR1 = R2/
945 FORMAT(1H0=5H)LBASE=7H LRPPO=7H LTRIP=7H NEW/5*917)
946 FORMAT(1H1=17X)ENTER=LEAVE TABLE/
947 FORMAT(1H0=11(2X=43)/11=
948 FORMAT(1H0=27X)ERROR IN DESCRIPTION OF TEC=5X/
  1 41XHEIGHT VECTOR IS PARALLEL TO BASE ELLIPS

C DATA ITY(1),ITY(2),ITY(3),ITY(4),ITY(5),ITY(6),ITY(7),ITY(8)/
  1 / 3HBOX,3HSPH,3HRCC,3HREC,3MTRC,3MELL,3MRAM,3MARB /
DATA ITY(9),ITY(10),ITY(11),
  1 / 3HTEC,3HTOR,3HARS /
DATA IAA(1),IAA(2),IAA(3),IAA(4),IAA(5),IAA(6),IAA(7),IAA(8)/
  1 / 2M,2HQ,2HR,2HA,2AM,2AR,2H A,2HA /
DATA IAN(1),IAN(2),IAN(3),IAN(4),IAN(5),IAN(6),IAN(7),IAN(8)/
  1 / 4,H,1,1,1,1,2,2,3,3 /
DATA IRL/1H /
C DO 10 I=1,11
10 NBODY(I)=0
C C ENTER AND PRINT OUT TITLE OF THE PROBLEM
WRITE (6,901)
READ(5,902)(IT(I),I=1,10)
WRITE (6,903)(IT(I),I=1,10)
C ENTER AND PRINT OUT THE PROGRAM CONTROL PARAMETERS
READ(5,904)NRPP,NTRIP,NSCAL,N800Y,NRMAX,IPHIN,IRCHEK
WRITE (6,905)NRPP,N800Y,NRMAX
C C RPP
WRITE (6,906)
LAR=1
C C RPP DATA INPUT
IF(NRPP.LE.0)GOTO 20
CALL RPPIN(LAR)
IF(IERR.GT.0)RETURN
C LBODY STORAGE RESERVE 3*(NRPP*N800Y) WORDS
  / BODY NUMBER / LOC OF POINTER TO BODY DATA/
  / REGION ENTER TABLE POINTER / REGION LEAVE TABLE POINTER /
  / NUM REGIONS IN ENTER TABLE / NUM REGIONS IN LEAVE TABLE /
20 L800Y=ND0=2

FIG. 75. (Contd.)
L=LAR
L=body=L+1
Ldata=L=body+3*(Nbody+NRP)
L=body=L=data

50 WRITE (6,911)

C ENTER DATA FOR BODY

C DO 370 N=1,Nbody
370 N=N+NRPP
LS1=0
READ (5,912) IC(1),IC(2),IC(3),ITYPE,IC(4),(FX(K),K=1,6)
DO 51 I=1,11
IF (ITYPE.EQ.ITY(1)) GOTO 52
51 CONTINUE
WRITE (6,913) ITYPE
STOP
52 ITYPE=I
NBOD(I)=NBOD(I)+1
K=L=body+3*(NRPP*N-1)
MASTER(K)=ITYPE=I15+LDATA

C BOX SPH RCC REC TRC ELL RAW ARB TEC TOR ARS

200 GOTO (201*220*202*201*203*202*201*230*204*203*240),ITYPE

201 L=12
GOTO 210
202 L=7
GOTO 210
203 L=8
GOTO 210
204 L=13
210 WRITE (6,915) N,IC(1),IC(2),IC(3),ITYPE,IC(4),(FX(J),J=1,6)
READ (5,940) (FX(J),J=7,LE)
WRITE (6,916) (FX(J),J=7,LE)

C BOX SPH RCC REC TRC ELL RAW ARB TEC TOR ARS

GOTO (290*300*300*290*280*270*290*300*260*250*300),ITYPE

C SPH

220 WRITE (6,915) N,IC(1),IC(2),IC(3),ITYPE,IC(4),(FX(J),J=1,4)
GOTO 300

C ENTER BODY DATA FOR AR

C 230 WRITE (6,915) N,IC(1),IC(2),IC(3),ITYPE,IC(4),(FX(J),J=1,6)
CALL ALBERT(FX,LBOT,NOQ,LS1)
GOTO 360

C ENTER BODY DATA FOR ARS

C 240 CALL ARIN(LBOT,L=DATA,M=ASTER,M=ASTER*M=H)
GOTO 360

C TOP CONVERT NORMAL VECTOR TO UNIT VECTOR

C 250 TT(1)=FX(4)
TT(2)=FX(5)
TT(3)=FX(6)
CALL UNIT(TT)
FX(4)=TT(1)
FX(5)=TT(2)
FX(6)=TT(3)
IF (FX(7).GE.FX(A(1))) GOTO 280
WRITE (6,913) IERR=IERR+1
GOTO 280

FIG. 75. (Contd.)
VERIFY SEMI-MAJOR AND SEMI-MINOR AXES PERPENDICULAR

260  FX(15) = FX(13)
     LE = 15
     TT(1) = FX(7)
     TT(2) = FX(8)
     TT(3) = FX(9)
     TT(1) = FX(10)
     TT(2) = FX(11)
     TT(3) = FX(12)
     IF (ABS DOT (TT) + TT2) LE + 0.01) GOTO 265
     WRITE (6,942) NN, TT1 + TT2
     IERR = IERR + 1

COMPUTE SEMI-MAJOR AXIS LENGTH AND CONVERT
SEMI-MAJOR AXIS TO UNIT VECTOR

265  FX(13) = SQRT DOT (TT + TT1)
     CALL UNIT (TT1)
     FX(10) = TT(1)
     FX(11) = TT(2)
     FX(12) = TT(3)

COMPUTE SEMI-MINOR AXIS LENGTH AND CONVERT
SEMI-MINOR AXIS TO UNIT VECTOR

266  FX(14) = SQRT DOT (TT2 + TT1)
     CALL CROSS (TT + TT2)
     CALL UNIT (TT)
     MDN = FX(4) * TT(1) + FX(5) * TT(2) + FX(6) * TT(3)
     TF (MDN) = 267 * 266 * 268
     WRITE (6,948)
     IERR = IERR + 1
     GOTO 268

267  TT(1) = TT(1)
     TT(2) = TT(2)
     TT(3) = TT(3)
     TT(7) = TT(1)
     TT(8) = TT(2)
     TT(9) = TT(3)
     GOTO 280

COMPUTE FOCI FOR ELL

270  IF (IC(4) EQ 8) GOTO 300
     ASQ = FX(4) * FX(4) + FX(5) * FX(5) + FX(6) * FX(6)
     C = SQRT (ASQ - FX(7) * FX(7))
     A = SQRT (ASQ)
     FX(7) = A * A

COMPUTE X+Y+Z COMPONENTS OF FOCI

271  CX = C * FX(4) / A
     CY = C * FX(5) / A
     CZ = C * FX(6) / A

COMPUTE X+Y+Z COORDINATES OF FOCI

272  FX(4) = FX(1) + CX
     FX(5) = FX(2) + CY
     FX(6) = FX(3) + CZ

FIG. 75. (Contd.)
FX(1)=FX(1)-CX
FX(2)=FX(2)-CY
FX(3)=FX(3)-CZ

C23 PRINT OUT NEW INPUT

C260 WRITE (6,915) NN, IC(1), IC(2), IC(3), ITY(ITYPE), IC(4), (FX(J), J=1,6)
WRITE (6,916) (FX(J), J=1,7) LE)
GOTO 300

C24 TRC VERIFY LOWER AND UPPER RADII NOT EQUAL

C285 IF (FX(7) .NE. FX(8)) GOTO 300
WRITE (6,944)
IERR=IERR+1
GOTO 300

C25 VERIFY THAT VECTORS ARE PERPENDICULAR IF BOX, RAW, OR REC

C290 IF (ABS(FX(4)*FX(7)+FX(5)*FX(8)+FX(6)*FX(9)), LE.0.01) GOTO 291
WRITE (6,942) NN, (FX(J)+J#9)
IERR=IERR+1
291 IF (ABS(FX(4)*FX(10)+FX(5)*FX(11)+FX(6)*FX(12)), LE.0.01) GOTO 292
WRITE (6,942) NN, (FX(J)+J#9)
IERR=IERR+1
292 IF (ABS(FX(7)*FX(10)+FX(8)*FX(11)+FX(9)*FX(12)), LE.0.01) GOTO 300
WRITE (6,942) NN, (FX(J)+J#9)
IERR=IERR+1

C26 STORE BODY DATA AND BODY DATA POINTERS IN MASTER=ASTER ARRAY

C30 GOTO (310*230*310*310*40+330*30+310*310*40+330*310*230+350*340*240)*ITYPE

C27 POINTER FORMAT BOX RAW / V / H1 / REC / V / M /
M2 H3 / N1 / M2 /

C310 CALL SE3(IWH,ASTER,MASTER,FX(1),FX(2),FX(3)*LBOT,DATA+NDQ+LS1)
MASTER(LDATA)=IWH#15
CALL SE3(IWH,ASTER,MASTER,FX(4),FX(5),FX(6)*LBOT,DATA+NDQ+LS1)
MASTER(LDATA)=MASTER(LDATA)+IWH
CALL SE3(IWH,ASTER,MASTER,FX(7),FX(8),FX(9)*LBOT,DATA+NDQ+LS1)
MASTER(LDATA)=IWH#15
CALL SE3(IWH,ASTER,MASTER,FX(10),FX(11),FX(12)*LBOT,DATA+NDQ+LS1)
MASTER(LDATA)=MASTER(LDATA)+IWH
DATA=DATA+2
GOTO 360

C28 POINTER FORMAT SPH / V / R /

C320 CALL SE3(IWH,ASTER,MASTER,FX(1),FX(2),FX(3),LBOT,DATA+NDQ+LS1)
MASTER(LDATA)=IWH#15
LS1=1
CALL SE3(IWH,ASTER,MASTER,FX(4),FX(5),FX(6)*LBOT,DATA+NDQ+LS1)
LS1=0
MASTER(LDATA)=MASTER(LDATA)+IWH
DATA=DATA+1
GOTO 360

FIG. 75. (Contd.)
FIG. 75. (Contd.)
C32  CHECK IF ANY MORE ROOM FOR SOLID DATA
C
360  IF(LDATA.LT.NDQ)GOTO 370
    WRITE (6,917) LDATA,LBOT,NDQ
    STOP
370  CONTINUE
    WRITE (6,918)
    WRITE (6,917) LDATA,LBOT,NDQ
    WRITE (6,945) LB, LRPPD, LABUT, LBDY, LBD, LDATA, LBOT, LSCALE, LTRIP
    NDQ
C33  TRANSFER (LBOT TO NDQ) TO ASTER (LDATA TO LDATA + LSUB)
C
LD=LDATA=1
LSUB=LBOT=LO=1
DO 375 I=LBOT,NDQ
    ASTER(LDATA)=ASTER(I)
LDATA=LDATA+1
375  CONTINUE
C34  UNPACK POINTER WORDS AND RECOMPUTE POINTERS TO
C     COMPENSATE FOR TRANSFER
C
INDEX=LBODY+3 *(NRPP+NBDY)
DO 390 I=INDEX,L
    CALL UN2(I+1,1,1,1,12)
    IF(I1.NE.0) I1=I1+LSUB
    IF(I2.NE.0) I2=I2+LSUB
    MASTER(I1)=I1*I15+12
390  CONTINUE
C35  REGION STORAGE
C
    WRITE (6,920)
    N=0
    J=0
    LREGD=LDATA
    LDATA=LDATA+NRMAX
    LREGD=LDATA
C36  ENTER REGION DATA
C
400  READ (5,921) IR,(IA(I),IN(I),I=1,9)
C37  CHECK VALIDITY OF REGION DATA
C
    DO 410 I=1,9
        IF(ABS(IN(I)).LE.NBDY*NRPP) GOTO 410
    WRITE (6,922) IR+1
    J=J+1
410  CONTINUE
C38  PACK AND STORE REGION POINTER DATA
C     LREGD / POINTER TO REGION / NUMBER OF BODIES IN REGION /
C
    IF(IR)440,420,421
420  WRITE (6,923) (IA(I),IN(I),I=1,9)
    GOTO 430
421  N=N+1

FIG. 75. (Contd.)
WRITE (6,924) IR,(IA(I),IN(I),I=1,9)
M=LRREG0+N+1
MASTER(M)=LDATA+I5

C C39 CHECK AND CONVERT OPERATOR TO NUMERICAL VALUE
C
430 DO 435 I=1,9
431 DO 433 K=1,8
432 IF(IA(I) .EQ. IAA(K)) GOTO 432
433 CONTINUE
434 WRITE (6,925) IA(I)+1
435 STOP

C C40 PACK AND STORE REGION DATA / OPERATOR / BODY NUMBER /
C
436 MASTER(LDATA)=IA(I)*I5+IN(I)
437 LDATA=LDATA+1
438 MASTER(M)=MASTER(M)+1
439 IF(LDATA .LT. NOD) GOTO 435
440 WRITE (6,927) LDATA+NOD
441 STOP

C C41 END REGION READ = TEST NUMBER OF REGIONS
C
442 IF(N .GE. NRMAX) GOTO 441
443 WRITE (6,926) IR
444 STOP

C C42 TEST FOR REGION CHECK OPTION, CHECK REGION DATA IF NOT ZERO
C (ERROR IF ANY POINT CAN BE IN MORE THAN ONE REGION)
C
446 IF(IRCHEK .EQ. NO) GOTO 500
447 WRITE (6,937)
448 LL=0
449 MIS=0

C
450 DO 456 I=1,NRMAX
451 J=I+1
452 DO 455 J=J+1,NRMAX
453 K=LRREG+I-1
454 CALL UN2(KRI+LOCNI+NUMI)
455 CALL UN2(KRI+LOCJ+NUMJ)
456 IF(NUMJ .GE. NUMI) GOTO 450
457 TO=NUMJ
458 TI=NUMI
459 GOTO 451
460 TI=NUMJ
461 TI=NUMI
462 L=LOCI

FIG. 75. (Contd.)
LOCJ=LOCJ+1
LOCJ=LOCJ+1

451 DO 453 K0=1,IO
KLK=LOCJ+K0-1
CALL UN2(KLK,IOP0,NBO)
DO 452 KI=1,II
KLK=LOCJ+KI-1
CALL UN2(KLK,IOP1,NBI)
IF(IPO0,.NE.,IOP1)GOTO 452
IF(NBO,.NE.,NBI)GOTO 452
MIS=MIS+1
GOTO 453

452 CONTINUE
453 CONTINUE
IF(MIS,.NE.,II)GOTO 454
WRITE (6,929) J,I
LL=LL+1
454 MIS=0
455 CONTINUE
WRITE (6,930) I
456 CONTINUE
IF(LL.GT.0)STOP
WRITE (6,937)

C43 PREPARE REGION LEAVE TABLE (IS=1) AND REGION ENTER TABLE (IS=1)

C

500 IS=1

NN=NBODY*NRPP
LENL=LDATA
DO 590 MMM=1,2
DO 580 N=1,NN
MM=LBODY+3*(N-1)
IF(IS.GE.0)GO TO 510
MASTER(M+1)=MASTER(M+1)+LDATA
GO TO 520
510 MASTER(M+1)=MASTER(M+1)+LDATA*I15

C

520 DO 570 J=1,NRMAX
ITEM=REGION(J-1)
CALL UN2(ITEM,LOC*NC)
CALL UN2(LOC,IOP,NUM)
DO 560 N=1,NC
MM=LOC+N-1
CALL UN2(MM,IOPER+NUM)
IF(NUM,.NE.,1)GOTO 560
IF(IOP,NE.,1)OR(IOP,NE.,5)GOTO 540
IF(IOPER,.GT.,4)GOTO 530
IF(IS=1)560,550,560
530 IF(IS=1)560,551,560
540 IF(IS,LT.0)GOTO 551
550 MASTER(M+2)=MASTER(M+2)+I15
GO TO 552
551 MASTER(M+2)=MASTER(M+2)+1
552 MASTER(LDATA)=J
LDATA=LDATA+1
IF(LDATA,LT,NDG)GOTO 570
WRITE (6,931) LDATA,NDG,MM,I
STOP
560 CONTINUE

FIG. 75. (Contd.)
CONTINUE
WRITE (6,938) MMM
IS=IS+2
CONTINUE
RESERVE SPACE FOR RTN STORAGE, ROUT STORAGE AND
SUBROUTINE 81 TEMPORARY STORAGE
L1=LOOA DATA=1
NN=NRPP+NBODY
LRIN=LOOA DATA+1
LROT=LRIN+NN
L10=LROT+NN
LEGEOM=L10+NN
WRITE (6,932) LEOmM
WRITE (6,919) LREGD, LREGL, LEVL, LRIN, LROT, L10, LEOmM
TEST REGION ENTER/LEAVE TABLE PRINT OPTION
IF (IENTLV.EQ.0) RETURN
WRITE (6,946)
NNR=NBODY+NRPP
PRINT OUT REGION ENTER/LEAVE TABLES
DO 600 M=1,NNR
LOC=LBODY+3*M (M=1)
LOC=LOC+1
CALL UN2 (LOC, LENT, LEAV)
LOC=LOC+1
CALL UN2 (LOC, NENT, NEAV)
J1=LENT
J2=LEAV
J2=LEAV
WRITE (6,933) N, J1, J2, (MASTER(K), K=J1, J2)
J1=LEAV
J2=LEAV
WRITE (6,934) N, J1, J2, (MASTER(K), K=J1, J2)
CONTINUE
TEST MASTER-ASTER ARRAY PRINT OPTION
IF (IPRIN.EQ.0) RETURN
WRITE (6,935)
PRINT OUT MASTER-ASTER ARRAY TO END OF REGION ENTER/LEAVE TABLES
DO 620 K=LBASE, L1, 3
IK=K
IK2=K+2
M=0
DO 610 I=IK+IK2
M=M+1
CALL UN2 (I, I, 12)
NO1(M)=I
NO2(M)=12
O4(M)=ASTER(I)
NO0(M)=I
CONTINUE
WRITE (6,936) (NO0(L), NO1(L), NO2(L), O4(L), L=1, 3)
FIG. 75. (Contd.)
FIG. 75. (Concluded)
SUBROUTINE RPPIN(LAR)
DIMENSION X(6)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/GEOM/LBASE,RIN,ROUT,LRI,LHO,PINF,IER,R,DIST
COMMON/UNGEOM/RRP,NTRIP,NSCALE,NGEOY,NRMXY,LTIRIP,LSCALE,LREGD+
1 DATA/LRI,LROT,LTO,LUCDA/I15/I30/LBODY/NASC,KLUOOP
COMMON/LRPP/LRPPL,LABUT
EQUIVALENCE(MASTER,ASTER)

C FORMAT(6E12.5)
C 920 FORMAT(18X+17F4.5) 35)
C 930 FORMAT(1H0+2HERROR IN DESCRIPTION OF RPP+5X+10HMIN-GE+MAX)
C 940 FORMAT(1H0+2HERROR IN DESCRIPTION OF RPP+7X+10X+110)
C 950 FORMAT(10X+7HDESCRIPTION OF RPP+5X+8X+2E20+6)
C
C IERR=0
N=1

C LRASE = BEGINNING LOCATION OF RPP POINTERS  RESERVE 12 WORDS/RPP
C / I / J / I (POINTER TO LIST OF ABUTTING RPP'S)
C / / K / J (NUMBER OF RPP'S THAT ABUT THIS SURFACE)
C K (POINTER TO BOUNDARY COORDINATE FOR SURFACE)

C I=LRASE+12*NRP

C LRPPD = BEGINNING LOCATION OF RPP BOUNDARY COORDINATES
C THAT ARE POINTED TO BY K (LRASE + 12 * NRPP)

C LRPPD=1

C ENTER BOUNDARY COORDINATES OF RPP

10 READ(5,910)(X(J)+J=1,6)
WRITE (6,920)N*{X(J)+J=1,6}

C VERIFY MINIMUM BOUNDARY COORDINATE LESS THAN CORRESPONDING
C MAXIMUM BOUNDARY COORDINATE

C DO 20 J=1,N+2
TF(X(J)+LT,X(J+1))GO TO 20
WRITE (6,930)N
STOP

20 CONTINUE

C STORE BOUNDARY COORDINATES BEGINNING AT
C LOCATION LRASE + (12 * NRPP)

C DO 30 J=1,6
II=LRBASE+12*NRP
L=LRBASE+12*(N=1)+2*(J=1)
30 IF(I+LJ*I)GO TO 31
ASTER(I)=X(J)
MASTER(I+1)=1
[I]=1
GO TO 33

C CHECK FOR AND ELIMINATE REDUNDANT BOUNDARY COORDINATES

31 IF(X(J)+1LJ,ASTER(I))GO TO 32
[I]=I+1
GO TO 30

FIG. 76. Source Listing, Subroutine RPPIN
FIG. 76. (Contd.)
IF(NRPP+LE.1) GOTO 63

TEST VALIDITY OF RPP DATA

DO 62 J=1,6
NRPP1=NRPPI
DO 61 I=1,NRPP1
JJ=LBASE+12*(I-1)+2*(J-1)
CALL UN2(JJ,INDUM,12)
I3=MASTER(JJ+1)
IF(I3+NE.0) GOTO 61
I3=I+1
DO 60 K=I1,NRPP
KK=LBASE+12*(K-1)+2*(J-1)
CALL UN2(KK,INDUM,15)
I6=MASTER(KK+1)
IF(I6+NE.0) GOTO 60
IF(I1+EQ.16) GOTO 60
IERR=IERR+1
WRITE (6+940) I+K
WRITE (6+950) J+ASTER(I3)+ASTER(I6)
CONTINUE
GOTO 62
61 CONTINUE
62 CONTINUE
63 LAR=L
RETURN
END

FIG. 76. (Concluded)
SUBROUTINE ALBERT (FX+LBOT+NOQ+LS1)
DIMENSION IA(6+4),AA(8+3)+F(4)+FX(6)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/UNCGEM/NRPP+TRIP+NSCAL+NBODY+NRMAX+LTRIP+LSCAL+LREGD+
   1 LDATA=LRTN+LRTOT+LLO+LLOCA+I5*I30+L8ODY+NASC+KLOOP
COMMON/GEOM/BLASE+RINT+ROUT+LRI+LRO+PIN+FERR+DIST
EQUIVALENCE (ASTER, MASTER)
C
901 FORMAT (25X+6F12.5)
902 FORMAT (10X+6(1X+4I1))
903 FORMAT (16X+6(1X+4I2))
904 FORMAT (1H0+12SHUNDEFINED PLANE)
905 FORMAT (15X+10E11.4)
906 FORMAT (15X+10E11.4)
907 FORMAT (1H0+25HFOUR POINTS NOT IN A PLANE)
908 FORMAT (1H0+25HERROR IN SIDE DESCRIPTION)
909 FORMAT (1H0+16HDEGENERATE PLANE; I5)
C
C1 STORE COORDINATES OF FIRST TWO VERTICES IN Array AA
C
K=1
DO 10 I=1,2
DO 10 J=1,3
AA(I,J)=FX(K)
K=K+1
10 CONTINUE
C
C7 ENTER COORDINATES OF REMAINING SIX VERTICES INTO Array AA
C
READ (5,903) (AA(I,J)+J=1,3),I=1,8
C
C3 ENTER ORDINAL NUMBERS OF PLANE VERTICES
C
READ (5,902) (IA(I,J)+J=1,4),I=1,6
WRITE (6+901) (AA(I,J)+J=1,3),I=1,8
WRITE (6+906) (IA(I,J)+J=1,4),I=1,6
C
DO 70 I=1,6
C
C6 RETRIEVE FIRST THREE VERTEX COORDINATES OF PLANE
C
IX=IA(I+1)
IY=IA(I+2)
IZ=IA(I+3)
XI=AA(IX+1)
YI=AA(IY+1)
ZI=AA(IZ+1)
X2=AA(IX+2)
Y2=AA(IY+2)
Z2=AA(IZ+2)
X3=AA(IX+3)
Y3=AA(IY+3)
Z3=AA(IZ+3)
C
C8 COMPUTE COEFFICIENTS OF PLANE EQUATION
C
D=X1*Y2*Z3+Z2*Y1*Z3+Z2*Y1*Z3+X1*Y3*Z2+Y1*Z2*Z1*Y2
A=(-Y2*Z3+Z2*Y3+Y1*Z3-Z1*Y3+Y1*Z2-Z1*Y2)
B=(X2*Z3-Z2*X3+X1*Z3+X3*Z1+X1*Z2-Z1*Y2)
C=(Y2*X3-X2*Y3-Y1*X3+Y1*X3+Y1*Z2-Z1*Y2)

FIG. 77. Source Listing, Subroutine ALBERT
D12=(X1-X3)**2+(Y1-Y3)**2+(Z1-Z3)**2

TEST FOR DEGENERATE PLANE

A2B2C2=A*A+B*B+C*C

IF(A2B2C2<NE.0) GOTO 21
WRITE (6,909) I
FORMAT (I0)
GOTO 61

TEST FOR UNDEFINED PLANE

21 D1210=D12*1.0E=12
IF(A2B2C2*GT*D1210) GOTO 22
WRITE (6,905)
FORMAT (I0)
IFERR=IFERR+1
GOTO 70

22 S=SQRT(A2B2C2)
WX=A/S
WY=B/S
WZ=C/S

RETRIEVE COORDINATES OF FOURTH VERTEX ON PLANE

IC=IA(I,4)
XA=AA(1C+1)
YA=AA(1C+2)
ZA=AA(1C+3)

COMPUTE DISTANCE TO PLANE OF FOURTH VERTEX

D2=((A*X4)+(B*Y4)+(C*Z4))/(A*X1+B*W1+C*Z1)

D2=0 D2

DETERMINE IF FOURTH VERTEX LIES ON PLANE OF FIRST THREE VERTICES

IF(D2*LE.1.01) GOTO 30
WRITE (6,907)
IFERR=IFERR+1
WRITE (6,906) I+A+B+C+D+D12+D2
GOTO 70

DO 31 K=1,4
F(K)=0
F(K)=0
K=K+1
31 CONTINUE

COMPUTE VALUES OF OTHER FOUR VERTEXES
WITH RESPECT TO PRESENT SIDE

L=1
DO 32 J=1,8
IF(J=EQ.IX.OR.J=EQ.IY.OR.J=EQ.IZ.OR.J=EQ.IC) GOTO 32
F(L)=A*AA(J+1)+B*AA(J+2)+C*AA(J+3)+D*AA(J+4)
L=L+1
32 CONTINUE

COMPUTE NUMBER OF OTHER VERTEXES ON EITHER
SIDE OF PLANE OR ON PLANE

FIG. 77. (Contd.)
M=0
N=0
J=0
DO 44 L=1,4
IF(Abs(F(L))·LE·1·0E-6) GOTO 42
IF(F(L))·41·42·43
41 M=M+1
GOTO 44
42 N=N+1
GOTO 44
43 J=J+1
44 CONTINUE

C13 DETERMINE SIDE OF PLANE OTHER VERTICES ARE LOCATED
C
IF(M·EQ·0) GOTO 51
IF(N·EQ·4) GOTO 60
IF(J·EQ·4) GOTO 61
GOTO 52
51 IF(M·EQ·4) GOTO 60
IF(J·EQ·4) GOTO 61
52 WRITE (6,906) I·A·B·C·D·D12·D2·(F(L)·L·1·4)
IERR=IERR+1
GOTO 70
C
C
C
60 A=A
B=B
C=C
D=D

C14 STORE PLANE COEFFICIENTS AND POINTERS
C
61 CALL SEE3(INH,MASTER,MASTER,A·B·C·LBOT·LODATA·NDQ·LS1)
MASTER(LODATA)=INH
LS1=1
CALL SEE3(INH,MASTER,D·D·LBOT·LODATA·NDQ·LS1)
LS1=0
MASTER(LODATA)=MASTER(LODATA)·INH·15
LODATA=LODATA+1
CONTINUE
RETURN
END

FIG. 77. (Concluded)
SUBROUTINE ARIN(LBOT, LDATA)

SUBROUTINE READS, CHECKS, PROCESSES, AND STORES INPUT DATA FOR THE ARS (ARBITRARY SURFACE)

DIMENSION W(3), UW(3), VW(3), WN(3)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/UNCLE/NN, IC(4)
EQUIVALENCE (MASTER, ASTER)

901 FORMAT(1H*I*(1,3A1),7X*3HARS, 2X*A4*2X, 8X*
1 37HNUMBER OF CURVES M=, 110 /
2 1H*33X*37HNUMBER OF POINTS PER CURVE N=, 110 /
3 1H*33X*37HNUMBER OF POINTS IN MN=, 110 /
4 1H*33X*37HNUMBER OF POINTS STORED NP=2N(M-1)=, 110 /
5 1H*33X*37HTOTAL STORAGE NSTR=4NP+82=, 110 )

903 FORMAT(25X,6F12.4)
904 FORMAT(10X,6F10.5)
905 FORMAT(1H*33X*34HNUMBER OF TRIANGLES DESCRIBED, 910 )
906 FORMAT(1H*33X*34HNUMBER OF NON-DEGENERATE TRIANGLES, 910 )
910 FORMAT(1H*0*4HERROR IN DESCRIPTION OF ARS SOLID NUMBER, 15)
911 FORMAT(5X,2HNUMBER OF POINTS IS 0)
920 FORMAT(10X,210C)

ENTER NUMBER OF CURVES AND NUMBER OF POINTS PER CURVE AND
COMPUTE NUMBER OF POINT TO BE STORED AND STORAGE REQUIREMENTS

READ(5,920) M,N
NN=M+N
NP=2*N*(N-1)
NSTR=4*NP+R2
WHITE(6,901) NN, IC(1), IC(2), IC(3), IC(4), M=N+MN, NP=NSTR

CHECK IF NUMBER OF POINTS IS 0

IF(NP.GT.0) GOTO 10
WHITE(6,910) NN
WHITE(6,911)
RETURN

RESERVE STORAGE IN MASTER-ASTER ARRAY FOR ARS DATA

LBOT=LBOT-NSTR
MASTER(LDATA)=LBOT
LDATA=LDATA+1
LOC=LBOT+82

ENTER AND STORE COORDINATE DATA OF ARS

LOC=LOC+4
DO 230 I=1, M
IF(I.EQ.M) LOC=LOC+4
L1=LOC
L2=LOC+4*(N-1)
READ(5,904) (ASTER(L), ASTER(L+1), ASTER(L+1), L=L1+L2+8)
WHITE(6, 903) (ASTER(L), ASTER(L+1), ASTER(L+1), L=L1+L2+8)
IF(I, NE, M) WRITE(6, 903)
IF(I, EQ, 1, 904) WRITE(6, 903)
IF(I, EQ, 1, NE, M) GOTO 220

FIG. 78. Source Listing, Subroutine ARIN
FIG. 78 (Concluded)
SUBROUTINE SEE3(IWH,ASTER,MASTER,FX,XX,XXX,LBOT,LDATA,NDQ,LS)

TEST TO DETERMINE IF TRIPLET OR SCALAR DATA
IF(LS,NE,0)GOTO 50
EXECUTE IF TRIPLET DATA
IF(LBOT,GT,NDQ)GOTO 20
NDQ=NDQ+2
SEARCH FOR EQUAL TRIPLET IN THE ASTER ARRAY
DO 10 I=LBOT,NDQ
IF(ASTER(I),NE,FX)GOTO 10
IF(ASTER(I+1),NE,FX)GOTO 10
IF(ASTER(I+2),NE,FX)GOTO 10
IWH=I
CONTINUE

STORE TRIPLET PASSED BY ARGUMENT LIST
20 ASTER(LBOT-1)=FX
ASTER(LBOT-2)=XX
ASTER(LBOT-3)=X
LWH=LWH-3
IF(LBOT,LE,LDATA)WRITE (6,30)LBOT,LDATA
RETURN

30 FORMAT(1H0,22HMEMORY OVERLAP IN SEE3,5X,5HMBOT=,I10,1
5X,5HLDATA=,I10)
EXECUTE IF SCALAR QUANTITY
SEARCH FOR EQUAL SCALAR QUANTITY IN THE ASTER ARRAY
50 DO 60 I=LBOT,NDQ
IF(ASTER(I),NE,FX)GOTO 60
IWH=I
RETURN

STORE SCALAR QUANTITY PASSED BY ARGUMENT LIST
60 CONTINUE
ASTER(LBOT-1)=FX
LWH=LWH-1
RETURN
END

FIG. 79. Source Listing, Subroutine SEE3
SUBROUTINE GRID

DIMENSION WP(3)

COMMON/PAREM/XB(3),WB(3),IR
COMMON/GEOM/LBASE,RIN,ROUT,LRO,PLN,F,IERR,DIST
COMMON/UNCGEN/NRP,NTL,P,NSCAL,NBODY,NMAXXLTR,NLSCALE,LREG,
LDATA,LRTN,LROT,LIO,LOCDATA,L3,LOBS,NSD,NSC,KLOOP
COMMON/ITRACK/D1,D2,KNRXXT,LMAX,TR(200),ABS(3),IRESTART,RENC

1 IT(200) 0,CA,CE,SA,SE
COMMON/CAL/NIR,SLO,ANGL,NTYPE,SPS,SC,LS(3),WS(3)
1 TRAVEYL/NV,IV,IVH
COMMON/WALTV/LRFO,NGM,ERR
COMMON/HOT/Y,REF,HEF
COMMON/CELL/CES,CEIL,
COMMON/CONTAL/ITESTG,IRAYS,IENTLY,IVOLUM,IONT,ITAPEB,NO,YES

C 901 FORMAT(8I10)
902 FORMAT(6EL12.8)
903 FORMAT(1H0,1H0,2HMN,15,SX,2MNX,15,SMX,7H1RSTT,15,SMX,15,ANR,15,SMX,
1 6HINSTT,15,SMX,15,ANR,15,SMX,15,SHI,15,SMX,15,SHI,15,SMX,
2 17H DATUM LINE AT Z=S F10.3+27M WITH RESPECT TO THE ORIGIN/
3 17H ORIGIN IS AT X=S F10.3+27M WITH RESPECT TO THE ORIGIN/
4 17H GROUND IS AT Y=S F10.3+27M WITH RESPECT TO THE ORIGIN/
5 17H GROUND IS AT Y=S F10.3+27M WITH RESPECT TO THE ORIGIN/
6 17H GROUND IS AT Y=S F10.3+27M WITH RESPECT TO THE ORIGIN/
7 17H GROUND IS AT Y=S F10.3+27M WITH RESPECT TO THE ORIGIN/

904 FORMAT(1H +7MINUTH=F12.5+5X+GHELEVATION=F12.5+5X+5X,
1 13M BACK OFF DIST=F12.5)
905 FORMAT(2E20.8+4E10.3)
907 FORMAT(1H0,1H0,15M CLERS SKIPPED)
908 FORMAT(1H0,42H OPTION SET TO COMPUTE RANDOM POINT IN CELL)
909 FORMAT(1H0,39H OPTION SET TO CHOOSE CENTER OF CELL)

C READ GRID INPUT PARAMETERS

READ(5,901)NX,NY,IRE,NRSTT,IRENC,NGM,ERR,RSTT,NEND,ICENT
READ(5,902)XSHIFT,YSHIFT,GROUND
READ(5,902)XSHIFT,YSHIFT,CELSIZ

C INITIALIZE PARAMETERS NOT SET BY INPUT

IF(IRESTT.LE.0)IRESTT=1
IF(CELSIZ.LE.0)CELSIZ=6
IF(NRSTT.LE.0)NRSTT=1
IF(NEND.NE.0)NEND=NX*NY
IF(NGM.ERR.LE.0)NGM.ERR=25

C PRINT OUT INPUT PARAMETERS

WRITE(6,903)NX,NY,NRSTT,IRENC,NRSTT,NEND,CELSIZ,
1 ZSHIFT,GROUND,XSHIFT,YSHIFT
WRITE(6,904)A,E,ENGRTH
IF(NWRT.EQ.1)WRITE(6,905)A,E,XSHIFT,YSHIFT,SHIFSTM,CELSIZ
WRITE(6,906)A,E,ENGRTH
IF(ICENT.EQ.0)WRITE(6,908)
IF(ICENT.NE.0)WRITE(6,909)
RADIAn=017453292519943
AR=A,RADIAn
ER=E,RADIAn
SA=SIN(AR)
CA=COS(AR)
SE=SIN(ER)
CE=COS(ER)

FIG. 80. Source Listing, Subroutine GRID
PROCESS NEND-NSSTART+1 CELLS

\[ \text{KK=NSSTART} \]

\[ \text{WH(1)=-CE*CA} \]
\[ \text{WH(2)=CE*SA} \]
\[ \text{WH(3)=-SE} \]

COMPUTE ROW AND COLUMN NUMBER OF GRID CELL

\[ \text{II=(KK-1)/NX+1} \]
\[ \text{JI=KK/NX} \]

\[ \text{CELL2=.5*CELSIZ} \]
\[ \text{V=FLOAT((NY/2)-II)*CELSIZ +CELL2} \]
\[ \text{VREF=V+CELL2} \]
\[ \text{H=FLOAT((NX/2)- JI)*CELSIZ +CELL2} \]
\[ \text{HREF=H+CELL2} \]
\[ \text{IF(ICENTR*EG+0)GOTO 5} \]
\[ \text{H=HREF} \]
\[ \text{V=VREF} \]
\[ \text{IV=H} \]
\[ \text{GOTO 6} \]

\[ \text{IV=RAN(-1)*10.} \]
\[ \text{IH=RAN(-1)*10.} \]
\[ \text{IVM=10*I+IV} \]

COMPUTE RANDOM POINT WITHIN GRID CELL

\[ \text{V=CELSIZ *FLOAT(IV)/10.*CELSIZ /20.} \]
\[ \text{H=CELSIZ *FLOAT(IV)/10.*CELSIZ /20.} \]

CONVERT GRID PLANE COORDINATES TO COORDINATES OF TARGET

\[ \text{XBS(1)=XSHIFT=V*CA+SE=H*SA} \]
\[ \text{XBS(2)=YSHIFT=V*SA+SE=H*CA} \]
\[ \text{XBS(3)=ZSHIFT=V*CE} \]
\[ \text{CALL TROPIC(WP)} \]
\[ \text{XBS(1)=XBS(1)+WP(1)*1.0E+4} \]
\[ \text{XBS(2)=XBS(2)+WP(2)*1.0E+4} \]
\[ \text{XBS(3)=XBS(3)+WP(3)*1.0E+4} \]

BACK OFF RAY ORIGIN FROM GRID PLANE TO ATTACK PLANE

\[ \text{XB(1)=XBS(1)-ENGLISHWARE} \]
\[ \text{XB(2)=XBS(2)-ENGLISHWARE} \]
\[ \text{XB(3)=XBS(3)-ENGLISHWARE} \]
\[ \text{IF(XB(3)+LF.GROUND)GOTO 40} \]

SAVE RAY ORIGIN AND DIRECTION COSINES OF RAY FOR LATER REFERENCE

\[ \text{DON 20 KK1=1} \]
\[ \text{XS(KK1)=XB(KK1)} \]
\[ \text{WS(KK1)=WB(KK1)} \]
\[ \text{CONTINUE} \]
\[ \text{CALL TRACK} \]
\[ \text{IF(1RAYS+E,NG1ERR)RETURN} \]
\[ \text{IF(1RAYS+E,NO)GOTO 40} \]

FIG. 80. (Contd.)
C10. COMPUTE RANDOM NUMBER OF CELLS (0-25) TO BE SKIPPED

 C
 MSHIFT=RAN(-1)*25.
 WRITE (6,907) MSHIFT
 KK=KK+MSHIFT
 KK=KK+1
 IF(KK.LE.NEND) GOTO 4
 RETURN
 END

FIG. 80. (Concluded)
SUBROUTINE TRACK
DIMENSION XP(3),ERROR(2)
COMMON/PAREM/XX(3),WB(3),IR
COMMON/EOEM/LBASE,MRN+OUTL+RI+LRO*LRP,FERR,ERR
COMMON/UNCGEM/RGBR+PRPP+TRIP+NSCL+NBODY+NRMAX+LSTRP+LSCALE+LRGE
1 LDAT+LRIN+LRIF+LIO+LOCA+I150+I30+LBODY+NASC+KLOOP
COMMON/GTRACK/O1+D2+KMAT,LMAX+TR(200),ABS(3),IRSR+IENC
1 IR(2000)+CA+CE+SA+SE
COMMON/CAL/NIR+SLS+SANGLE+NTYPE+SSPACE+LXS(3),WS(3),TRAVEL
1 SNLEV+SIV
COMMON/CONT/RITTEST+IRAYSK+IENTRY+VOLUM+IWO+ITAPAG+NO+IYES
COMMON/WALT/LWFO+NW1
COMMON/HOT+VREF+HREF
COMMON/LSU/LSURF
COMMON/CELL/CESIZ
COMMON/ERR/IERR
901 FORMAT(F6.1,1X,F6.1,1X,F7.2,1X,F7.2,4I2,13,1X,F213,1X,F8.3)
902 FORMAT(14,F7.2,F7.2,F6.1,F7.2,F8.3,F7.2,F213,1X,F13,1X,F13,1X,F13,1X,F13,1X,F13)
903 FORMAT(11H NUMBER OF INTERSECTIONS,GT,200)
904 FORMAT(/)
905 FORMAT(16H0 ITEM IN CELL (+I4+1H++,I4+2H2++I4+1H+I5+)
1 2HV*1F6.1,1X,F8.3+) .
C
C ERROR(2)= 6H0 ITEM
C DATA ERROR(1),ERROR(2)/4H ,4H ITEM/
C I12=0096
C NASC=1
C ITR=IRSTAT
C L=1
C KHID=0
C JCNT=0
C MSKAT=0
C MTARG=1
C MARM=0
C MVOL=0
C
C DO 10 I=1,200
C ITR(I)=0
C TR(I)=0,
C 10 CONTINUE
C
C SUBROUTINE G1 WILL RETURN WITH S1=DISTANCE THRU REGION IR,
C IRPRIM=THE NEXT REGION NUMBER, XP=INTERSECT OF NEXT REGION
C
C 20 CALL G1(S1,IRPRIM,XP)
C IF(IRPRIM.LT.0)RETURN
C TR(L)=S1
C KLsurf=Lsurf+7
C LOC=LIRFO+IR=1
C CALL UN2(LOC,DUM,IDENT)
C IDENT=IDENT+1
C
C PACK SURFACE NUMBER = BODY NUMBER = NEXT REGION NUMBER
C
C ITR(L)=(KLsurf*112+NASC)*I12+IRPRIM
C IF(NASC.LE.NRPP)IRPRIM=0

FIG. 81. Source Listing, Subroutine TRACK
IF(IPNPRMT,EG,+0)GOTO 100
IF(=IPNPRMT
KHIT=KMIT+1
IF(L*GT+1)GOTO 40

COMPUTE DISTANCE FROM GRID PLANE TO FIRST INTERSECT OF TARGET
D1=-(XP(1)-XBS(1))*WS(1) + (XP(2)-XBS(2))*WS(2)
   + (XP(3)-XBS(3))*WS(3))
GOTO 60

TEST SPACE IDENTIFICATION CODE
0 = NO SPECIAL MATERIAL 10=SKIRT 20=ARMOR 30=TARGET
   -12=91=19=29=...=91=99 = INTERIOR VOLUME
   1 = EXTERIOR VOLUME
40 IF(IDENT,EG,+0)GOTO 60
   IF(IDENT=IDENf/10)*10+EG,+0)GOTO 50
   KHIT=KMIT+1
   IF(IDENT,NE,+1)MVOL=1
   GOTO 60
50 IF(IDENT,EG,+20)MARR=1
   IF(IDENT,EG,+30)MTARG=1
   IF(IDENT,EG,+10)MSKRT=1
60 L=L+1
   IF(L*LE,+20)GOTO 20
WRITE (6+,903)
STOP

END OF RAY OUTPUT RESULTS
100 IF(L*EQ+1)RETURN
   IF(ITAPE8,EG,NO+AND.IWOT,EG,NO)RETURN

COMPUTE DISTANCE FROM GRID PLANE TO LAST INTERSECT OF TARGET
D2=XDIST(XBS,XP)+S1
D2=0
   IF(KHIT,GT,0)GOTO 105
   KHIT=KMIT+1
   MTARG=0
105 KHIT=KMIT+1
   IH=ABS(M/CELSIZ)*+5
   IF(M-LT,0)IH=IH
   IV=ABS(V/CELSIZ)*+5
   IF(V-LT,0)IV=IV

OUTPUT GRID CELL AND TARGET IDENTIFICATION DATA
IF(ITAPE8,EG,NO)GOTO 110
WRITE (6+,904)
WRITE (6+,901)HREF,VREF,IVIM,D1,D2,MSKRT,MTARG,MARR,MVOL,
1 KHIT,IM,IV+H
110 IF(IWOT,EG,YES)WRITE (1+,901)HREF,VREF,IVIM,D1,D2,MSKRT,MTARG,
1 MARR,MVOL,KHIT,IM,IV+H

FIG. 81. (Contd.)
OUTPUT RAY INTERSECTION DATA

LMAX=L
LM=L+1
TRAVEL=TR(1)
DO 200 KIK=1,LMAX+2
JERR=1
L=LM+1
IF(L.GE.LMAX)RETURN

COMPUTE DATA OUTPUT FOR FIRST HALF OF LINE

CALL CALC
IF(NIR,NE,0)GOTO 113
JERR=2
IERR=IERR+1
113 IF(SSPACE,NE,0)JCNT=JCNT+1

SAVE DATA OUTPUT FOR FIRST HALF OF LINE

NIR=NIR
SLOS=SLOS
ANGLE=ANGLE
SN=SN
NTYPE=NTYPE
SPACE=SSPACE

L=L+1
IF(L.LT.LMAX)GOTO 115
NIR=0
SLOS=0
ANGLE=0
SN=0
NTYPE=0
SSPACE=0
GOTO 120

COMPUTE DATE OUTPUT FOR SECOND HALF OF LINE

CALL CALC
115 IF(NIR,NE,0)GOTO 117
JERR=2
IERR=IERR+1
117 IF(SSPACE.EQ.0)GOTO 130
120 JCNT=JCNT+1
130 N=L-JCNT

TEST TRACK FLAG

501 = TRACK EDGE 502 = TRACK FACE
IF NORMAL DISTANCE 10 INCHES RAY ENTERS TRACK FACE

IF(NIR1,NE,501)GOTO 140
IF(SN1.LT.10.*NIR1=502
140 IF(NIR,NE,501)GOTO 150
IF(SN1.LT.10.*NIR=502

OUTPUT RAY INTERSECTION DATA

FIG. 81. (Contd.)
150 IF(IWOT.EQ.IYES) WRITE(1,902) NIR1, SLOS1, SN1, ANGLE1, NTYPE1, SPACE1
 1 NIR, SLOS, SN, ANGLE, NTYPE, SPACE, IH, IV = N
 IF(ITAPE8.EQ.IYES) WRITE(6,902) NIR1, SLOS1, SN1, ANGLE1, NTYPE1, SPACE1
 1 NIR, SLOS, SN, ANGLE, NTYPE, SPACE, IH, IV = N + ERROR(JERR0)
 IF(ITAPE8.EQ.NO .AND. JERR0.EQ.2) WRITE(6,905) IH, IV*HREF, VREF

C
 IF(L.GE.LMAX) RETURN
 IF(NTYPE .EQ. 9) RETURN
 CONTINUE
 RETURN
 END

FIG. 81. (Concluded)
SUBROUTINE CALC

DIMENSION XP(3), TEMP(3), TEM1(3), TEM(3), XHID(3), LENP(4),
     WN(3), WI(3), WA(3), XI(3), HF(3), VF(3)

DIMENSION MASTER(10000)
COMMON MASTER(10000)
COMMON MASTER(10000)
COMMON/GEOM/LBASE,RIN,ROUT,LRIP,LRO,PINF,ERR,DISP
COMMON/UNCOMP/HBASE,NSCAL,NBODY, NRMAX, LTRIP, LSCALE, LREGO
COMMON/STRACK/D01,D2,KHIT,LMAX,TRIC(200)*XS(3),XR(3), NCOL, IENC
COMMON/CAL/NIR,LSPACE,ANGLE, NTY,C, SPACE,L,XS(3), N(T)

REAL NRT(3)

C 901 FORMAT(1HO+15HTHATS ALL FOLKS//)
C 902 FORMAT(1HO+1MBAD ITYPE IN CALC.+5X+6ITYPE=+5X+6HNB0=+5X+6H/)
1 16H RETURN TO TRACK//
C 903 FORMAT(1HO+1NLSURF 4DARTS DID NOT FIND NORMAL+)
C 904 FORMAT(15H,NORM=5H,NIR=+5X+6ITYPE=+5X+6HNB0=+5X+6H,
1 6HLSURF=+10/4M WB=+3E20/10/4M X=+3E20/10/4M X3=+3E20/8/)
C 905 FORMAT(13H ERROR IN CALC. A TRC HAS R1 = R2 )
C 906 FORMAT(42H ERROR IN CALC. BAD LSURF FOR BOX OR RAW )

C1 RETRIEVE FOR PRESENT INTERSECT THE SURFACE NUMBER,
     BODY NUMBER, AND NEXT REGION

CALL OPENK,L,LSURF,NB0,NIR)
IF(NIR,GT,0)GOTO 10
WRITE(6,901)
RETURN

C2 COMPUTE TRAVEL = LINE-OF-SIGHT DISTANCE TO THIS REGION
     SLOS = LINE-OF-SIGHT DISTANCE THROUGH THIS REGION
     XI = COORDINATES OF INTERSECT POINT

10 SLOS=TR(L(*1))
   DO 20 I=1,3
      XI(I)=XS(I)+TRAVEL*WS(I)
   CONTINUE
   TRAVEL=TRAVEL+SLOS
   LSRF=LSURF+7

C3 SET THE CONSTANT MULTIPLIER OF THE DIRECTION COSINES OF NORMAL
     TO +1 FOR ENTRY OR -1 FOR EXIT

   XNOS=1.
   IF(LSURF,LT,0)*XNOS=-1.

C4 RETRIEVE BODY TYPE AND LOCATION OF DATA FOR INTERSECTED BODY

   LOC=LBODY(+3#(NB0=1)
   CALL UN21LOC,LT,ITYPE,LDATA)
   LSRF=1ABS(LSURF)
   ITYPE=ITYPE+1
   IF(ITYPE,GE,1 AND ITYPE,LE,15)GOTO 10

FIG. 82. Source Listing, Subroutine CALC
WRITE (6,902) ITYPE, NBO
RETURN

TRANSFER TO SPECIFIC BODY SECTION TO COMPUTE DIRECTION COSINES OF NORMAL

RPP BOX SPH REC TRC ELL RAW ARB TEC TOR ARS
30 GOTO (50, 100, 150, 200, 300, 350, 400, 450, 500, 550, 600), ITYPE

CHECK THE SPACE CODE AND ITEM CODE OF THE NEXT REGION

40 CALL OPENK(L*, DUM, DUM, NEXREG)
   ISPOT = LIRFO + NEXREG - 1
   CALL UN2(ISPOT, DUM, IDENT)
   ISPOT = LIRFO + NIR - 1
   CALL UN2(ISPOT, NIR + DUM)
   IDENT = IDENT - 1
   IF (IDENT = (IDENT / 10) * 10, NE.0) GOTO 41
   NTYE = IDENT
   SSPACE = 0.
   RETURN

41 L = L + 1
   IF (L*, LT = LMAX) GOTO 42
   IDENT = 0
   SSPACE = 0.
   NTYE = IDENT
   RETURN

42 NTYE = IDENT
   SSPACE = TR(L*)
   TRAVEL = TRAVEL + SSPACE
   RETURN

APP SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL

50 WR(1) = 0.0
   WR(2) = 0.0
   WR(3) = 0.0
   GOTO (51, 52, 53, 54, 55, 56) + LSURF
51 WR(1) = XNOS
   GOTO 1000
52 WR(1) = -XNOS
   GOTO 1000
53 WR(2) = XNOS
   GOTO 1000
54 WR(2) = -XNOS
   GOTO 1000
55 WR(3) = XNOS
   GOTO 1000
56 WR(3) = -XNOS
   GOTO 1000

FIG. 82. (Contd.)
GOTO 1000

C  BOX SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL

C 100 CONTINUE
   KCOM=LSURF=(LSURF/2)*2
   IF(KCOM.EQ.0)XNOS=NXNOS
   IF(LSURF.>3)104+103+105
   I=1
   GOTO 110
104 I=2
   GOTO 110
105 IF(LSURF.<LT.5)GOTO 103
   I=3
110 CALL UN2(LDATA+IEMP(4)+IEMP(1))
   LDATA=LDATA+
   CALL UN2(LDATA+IEMP(2)+IEMP(3))
   DO 115 J=1+3
      LW=IEMP(I)
      LV=IEMP(A)
      M=J-1
      IJK=LW+M
      IJK1=LV+M
      TEMP(J)=ASTER(IJK)+ASTER(IJK1)
      MK=J-1+IEMP(A)
      TEMP(J)=ASTER(MK)
   115 CONTINUE
   CALL DCOSEP(TEMP1+TEMP1*WB)
   DO 120 J=1+3
      WA(J)=XNOS*WB(J)
   120 CONTINUE
   GOTO 1000

C  SPH SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL

C 150 CALL UN2(LDATA,LV,DUM)
   DO 160 I=1+3
      MM=1+LV
      TEM(M)=ASTER(M)
   160 CONTINUE
   CALL DCOSEP(XI*TEMP,WB)
   DO 170 I=1+3
      WA(I)=XNOS*WB(I)
   170 CONTINUE
   GOTO 1000

C  RCC AND REC SECTION FOR COMPUTING DIRECTION COSINES OF NORMAL

C  FOR AN INTERSECT WITH EITHER PLANAR SURFACE

C 200 IF(LSURF<>2)203,201,210
201 XNOS=NXNOS
202 CALL UN2(LDATA+LV1+LV2)
   DO 203 I=1+3
      MM=1
      IJK1=MM+LV1
      IJK2=MM+LV2
      TEM(M)=ASTER(IJK1)
      TEM1(I)=ASTER(IJK1)+ASTER(IJK2)
   203 CONTINUE
   CALL DCOSEP(TEM1,TEM1*WB)

FIG. 82. (Contd.)
DO 204 I=1,3
WB(I)=XNOS*WB(I)
204 CONTINUE
GOTO 1000
C
C11 RCC AND REC SECTION FOR PROJECTING INTERSECT ONTO HEIGHT VECTOR
FROM THE QUADRATIC SURFACE
C
210 CALL UN2(LDATA+LV+LH)
LR1=MASTER(LDATA+1)
DO 211 J=1,3
M=J
IJK=LV*M
TEM(J)=ASTER(IJK)
IJK1=LM*M
TEM1(J)=ASTER(IJK)+ASTER(IJK1)
211 CONTINUE
CALL DCOSP(TEM+XI+WN)
CALL DCOSP(TEM+TEM+WI)
SUM=0.
DO 212 J=1,3
SUM=SUM+WN(J)*WI(J)
212 CONTINUE
DO 214 J=1,3
XP(J)=SUM*XDIST(TEM+XI)
XP(J)=XP(J)*WI(J)+TEM(J)
214 CONTINUE
C12 TRANSFER TO REC SECTION TO COMPUTE DIRECTION COSINES OF NORMAL
IF AN INTERSECT ON THE QUADRATIC SURFACE OF AN REC
C
C13 COMPUTE THE DIRECTION COSINES OF THE NORMAL IF AN INTERSECT ON
QUADRATIC SURFACE OF AN RCC
C
CALL DCOSP(XI,XP,WH)
DO 220 J=1,3
WH(J)=XNOS*WB(J)
220 CONTINUE
GOTO 1000
C
C14 COMPUTE THE DIRECTION COSINES OF THE NORMAL IF AN INTERSECT ON
QUADRATIC SURFACE OF AN REC
C
250 LDATA=LDATA+1
CALL UN2(LDATA+LR1+LR2)
DO 255 J=1,3
M=J
IJK=LR1*M
TEM(J)=ASTER(IJK1)+XP(J)
IJK=LR2*M
TEM1(J)=ASTER(IJK2)*XP(J)
255 CONTINUE
A1=XDIST(XP,TEM1)
A2=XDIST(XP,TEM)
IF(A1.0.EQ.A2)GOTO 260
A3=A1
A1=A2
A2=A3

FIG. 82. (Contd.)
TEMP(1) = TEMP(1)
TEMP(2) = TEMP(2)
TEMP(3) = TEMP(3)

260 C = SQRT(A1*A1 + A2*A2)
CALL DCOSP(XP, TEMP, WN)
DO 265 J = 1, 3
   TEMP(J) = XP(J) * C * WN(J)
   TEMP(J) = XP(J) * C * WN(J)
   CONTINUE
   CALL DCOSP(TEM, XI, WN)
   DO 270 J = 1, 3
      TEMP(J) = 2 * A1 * WN(J) * TEMP(J)
      CONTINUE
      CALL DCOSP(TEM, TEM, WB)
      DO 275 J = 1, 3
         TEMP(J) = 2 * A1 * WN(J) * TEMP(J)
      CONTINUE
      GOTO 1000

C

C 15 TRC SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL

C

300 CALL UN(2(LDATA + LV, LH))
   WN(1) = ASTER(LH)
   WN(2) = ASTER(LH + 1)
   WN(3) = ASTER(LH + 2)
   IF (LSURF, EQ, 3) GOTO 310
   IF (LSURF, EQ, 2) XNOS = XNOS
   CALL UNIT(WN)
   WH(1) = XNOS * WN(1)
   WH(2) = XNOS * WN(2)
   WH(3) = XNOS * WN(3)
   GOTO 1000

310 LDATA = LDATA + 1
   CALL UN(2(LDATA + LR1, LR2))
   RH = ASTER(LR1)
   RT = ASTER(LR2)
   RATIO = RH / (RH - RT)
   TEMP(1) = ASTER(LV)
   TEMP(2) = ASTER(LV + 1)
   TEMP(3) = ASTER(LV + 2)
   DO 320 I = 1, 3
      TEMP(I) = TEMP(I) * RATIO + WN(I) * XI(I)
      TEMP(I) = TEMP(I) - XI(I)
      CONTINUE
      CALL CROSS(WA, TEMP, TEM)
      CALL CROSS(WB, WA, TEM)
      CALL UNIT(WB)
      WH(1) = XNOS * WB(1)
      WH(2) = XNOS * WB(2)
      WH(3) = XNOS * WB(3)
      GOTO 1000

FIG. 82. (Contd.)
SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL

CALL UN2(LDATA+LR1+LR2)
L5=MASTER(LDATA+1)
DO 352 J=1+3
M=J+1
IJK1=M+LR1
IJK2=M+LR2
TEM(J)=MASTER(IJK1)
TEM1(J)=MASTER(IJK2)
352 CONTINUE
A=MASTER(J)
CALL DCOSP(TEM+XI+WN)
DO 353 J=1+3
TEM(J)*=WN(J)*TEM(J)
353 CONTINUE
CALL DCOSP(TEM,TEM+MP)
DO 354 J=1+3
WN(J)*=XNOS+WN(J)
354 CONTINUE
GOTO 1000

SECTION FOR COMPUTING THE DIRECTION COSINES OF NORMAL TO SLANTED SURFACE

IF(LSURF.EQ.2)GOTO 401
TRANSFER TO HX SECTION IF INTERSECT NOT ON SLANT SIDE

IF(LSURF.NE.4)GOTO 1000
WHITE (6+906)
STOP
CALL UN2(LDATA+LV+LV1)
LDATA=LUATA+1
CALL UN2(LDATA+LV2+LV1)
DO 410 J=1+3
M=J+1
IJK1=M+LV1
IJK2=M+LV2
TEMP(J)=ASTER(IJK1)
XMIN(J)=ASTER(IJK1)-ASTER(IJK2)
IJK3=M+LV3
TFM(J)=ASTER(IJK3)
410 CONTINUE
T=1
J=2
K=3
L=4
DO 411 KK=1+3
TFM1(I)=XMIN(J)*TEM(K)-XMIN(K)*TEM(J)
LK=1
L=K
J=L
K=LK
411 CONTINUE

FIG. 82. (Contd.)
SUM=0
DO 412 J=1,3
SUM=SUM+TEM1(J)*TEMP(J)*SUM
412 CONTINUE
SUM=SUM/AWS(SUM)
TLK=TEM1(1)**2+TEM1(2)**2+TEM1(3)**2
TLK=SRT(TLK)
DO 420 J=1,3
WH(J)=WHOS(SUM)*TEM1(J)/TLK
420 CONTINUE
GOTO 1000

C
C 19 ANH SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL
C
450 LSPT=LOADATA+LSURF=1
CALL UN2(LSPT,DOM+L1)
SUM=0
DO 451 J=1,3
SUM=SUM+ASTER(IJK)**2
451 CONTINUE
DIV=SRT(SUM)
DO 460 J=1,3
ASTER(IJK)=ASTER(IJK)/DIV
460 CONTINUE
GOTO 1000

C
C 20 TEC SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL
C
500 CALL UN2(LOADATA, LV, LH)
LOADATA=LOADATA+1
CALL UN2(LOADATA, LN, LA)
WN(1)=ASTER(LN)
WN(2)=ASTER(LN+1)
WN(3)=ASTER(LN+2)
IF(LSURF=2)520*510*530
510 XNOS=XNOS
520 WB(J)=XNOS*WN(1)
WB(J)=XNOS*WN(2)
W(J)=XNOS*WN(3)
GOTO 1000
530 LOADATA=LOADATA+1
CALL UN2(LOADATA, LR1+LR2)
LR2=ASTER(LOADATA+1)
VF(1)=ASTER(LV)
VF(2)=ASTER(LV+1)
VF(3)=ASTER(LV+2)
HF(1)=ASTER(LH)
HF(2)=ASTER(LH+1)
HF(3)=ASTER(LH+2)
TEMP(1)=XI(1)-VF(1)
TEMP(2)=XI(2)-VF(2)
TEMP(3)=XI(3)-VF(3)
HH=DOT(TEMP, W)
HDN=DOT(HF, W)
GAMMA=HH*HDN

FIG. 82. (Contd.)
TEMP(1)=VF(1)+GAMMA*HF(1)  
TEMP(2)=VF(2)+GAMMA*HF(2)  
TEMP(3)=VF(3)+GAMMA*HF(3)  
R1=ASTER(LR1)  
R2=ASTER(LR2)  
TAU=(R1/R2)**2  
R4=R2/ASTER(LR3)  
BSQ=(GAMMA*R4+R2*(1-GAMMA))**2  
ASQ=TAU*BSQ  
C=SQRT(ASQ-BSQ)  
TWOA=2*SQR(TAUS)  
DO 540 I=1,3  
IJK=LA+I-1  
TEMP1(I)=C*ASTER(IJK)  
TEM(I)=TEMP(I)+TEMP1(I)  
TEM1(I)=TEMP(I)-TEMP1(I)  
540 CONTINUE  
CALL DO CSP(TEM, XI, WN)  
TEMP(1)=TEM(1)+TWOA*WN(1)  
TEMP(2)=TEM(2)+TWOA*WN(2)  
TEMP(3)=TEM(3)+TWOA*WN(3)  
CALL DO CSP(TEM, TEM1, WN)  
IF(R2.EQ.R4) GOTO 545  
RATIO=R2/(R2-R4)  
HF(1)=VF(1)+RATIO*HF(1)-XI(1)  
HF(2)=VF(2)+RATIO*HF(2)-XI(2)  
HF(3)=VF(3)+RATIO*HF(3)-XI(3)  
545 CALL CROSS(NF, HF, WN)  
CALL CROSS(WB, NF, HF)  
CALL UNIT(WB)  
WH(1)=XNOS*WB(1)  
WH(2)=XNOS*WB(2)  
WH(3)=XNOS*WB(3)  
GOTO 1000  
C  
C21 FOR SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL  
C  
550 CALL UN2(LDATA, LV, LN)  
LDATA=LDATA+1  
CALL UN2(LDATA, LR1, DUM)  
DN 551 I=1,3  
J=I-1  
IJK=LV+J  
TEMP(I)=XI(I)-ASTER(IJK)  
IJK=LN+J  
TEMP(I)=ASTER(IJK)  
551 CONTINUE  
R1=ASTER(LR1)  
CALL CROSS(TEM, TEMP1, TEMP)  
CALL CROSS(TEM1, TEM, TEMP1)  
CALL UNIT(TEM1)  
FIG. 82. (Contd.)
DO 552 I=1,3
J=J-1
IJK=LV+J
TEMP(I)=ASTER(IJK)
TEMP1(I)=TEMP(I)+RI*TE(I)
552 CONTINUE
CALL OCOSP(TEMP1,XI,WH)
DO 553 I=1,3
WH(I)=XNOS*WH(I)
553 CONTINUE
GOTO 1000
C
C 22 A PS SECTION FOR COMPUTING THE DIRECTION COSINES OF THE NORMAL
C
DO 600 LOCARS=MASTER(LDATA)
LOC=LOCARS+2
DIS=XDIST(XS,XI)
DO 610 I=1,20
IF (ABS(DIS-ASTER(LOC)),LE,0.0001) GOTO 620
LOC=LOC+4
610 CONTINUE
WRITE(6,903)
SN=-1.
ANGLE=1.
RETURN
620 WH(1)=ASTER(LOC+1)
WH(2)=ASTER(LOC+2)
WH(3)=ASTER(LOC+3)
GOTO 1000
C
C 33 COMPUTE OBLIQUITY ANGLE AND NORMAL DISTANCE TO NEXT REGION
C
1000 DO 1001 J=1,3
XM(J)=XI(J)+WS(J)*1.0E=3
1001 CONTINUE
ANGLE=0.
DO 1002 J=1,3
ANGLE=ANGLE+WB(J)+WS(J)
1002 CONTINUE
IF (ABS(ANGLE),LE,1.0) GOTO 1010
ANGLE=0.
SN=0.
WRITE(6,904)NIR,ITYPE,NBO,LSURF,WB,WS,XP,XB,XI,XNOS
IR=NIR
GOTO 40
C
C 4 COMPUTE OBLIQUITY ANGLE
C
1010 ANGLE=TAN2(SORT(1,-ANGLE+ANGLE))AR(180.,3.,141592654)
IF (ANGLE,LE,0.0) GOTO 1020
DO 1011 J=1,3
WR(J)=WB(J)
1011 CONTINUE
GOTO 1000

FIG. 82. (Contd.)
FIG. 82. (Concluded)
SUBROUTINE Gl(S!, IRPRIM, XP)
DIMENSION XP(3), LSURT(50), NASCt(50)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PAREM/XB(3) * NB(3) * IR
COMMON/GEOM/LBASE,RIN,RROUT,LRI,LR0,PINF,IERR,DISt
COMMON/UNCEGM/NRRP,NTRIP,NTRIP0,NASCt,NASCt0,BODY(50)
1 LDATA,LRI,LROT,LT0,LOCDNS(150),L0B0DY,NASC+KLOOP
COMMON/CAL/NIR,SLOS,ANGLE,NTYPE,SSPACE,LS(3),WS(3),TRAVEL
1 SN=V0=IVIH
COMMON/WALT,LTRFO,NGTERR
COMMON/LSU/LSURF
COMMON/CTRL/TEST,IRAYSK,IENTLY,IVOLUM,ITOT,ITAPE,NO,IYES
COMMON/DAVIS/GRID,LOOP,INORM
COMMON/CELL/CELSIZ
COMMON/WHICH/NBO
EQUIVALENCE (ASTER+MASTER)
C
901 FORMAT(1HO,32HIIERROR IN GI AT 140 BAD I'"3MPE 5X,4H MITY,6I5)
902 FORMAT(1HO,33HIIIERROR IN GI AT 510 SM % PINF,5X,3H MIR,6I5)
903 FORMAT(4H XA=3E20, 8/6H WB=3E20, 8/10X,5XKLOOP=12X,3HNB0,
1 12X,3HLRI,12X,3HLRO,11X,4HNM1T=11X,4HLOOP=61I5)
904 FORMAT(1H1,15(2H, 3X, 9H ERROR NO.15,3X,1I5)
905 FORMAT(3X,4HCELL,2I4)
906 FORMAT(19H ERROR IN GI AT 640//4H J1=110+AH, J2=110+7HM LSURF=,
1 110+6H NASC=110+4H I1=110+4H SM=E21,10+4H S1=1E17+10/
2 AH WB=3E21,10/4H XA=3E21,10) 907 FORMAT(50H THE (SOLID POSITION/DEPTH/POINT NOW AT) IS ONE OF,
1 10 6H THESE/6H X A=3E21,10//6H DIST=,E21,10//)
908 FORMAT(6X,3HRTX,12X,4HROUT,7X,3H ENTERING,2X,3H LEAVING,3X,
1 8H BODY NO.,3X,3HRAY/=35X,8HSIDE NO.,2X,8HSIDE NO.,//)
910 FORMAT(/16H TILT RIN=ROUT=,E20+10+30X,2HI,6I5)
911 FORMAT(2(2X,E15.8),8X+12+8X+12+6A+15+5X,7MSTARTE,)
912 FORMAT(2(2X,E15.8),8X+12+8X+12+6A+15+5X,7MAS HIT,)
913 FORMAT(2(2X,E15.8),8X+12+8X+12+6A+15+5X,7LEAVE,)
914 FORMAT(2(2X,E15.8),8X+12+8X+12+6A+15+5X,7IN,)
915 FORMAT(2(2X,E15.8),8X+12+8X+12+6A+15+5X,7MENTERING,)
916 FORMAT(2(2X,E15.8),8X+12+8X+12+6A+15+5X,7MILL HIT,)
917 FORMAT(/4I14H END ERROR NO.14+3X,)
918 FORMAT(1HO,15+2I1H ERRORS IN GI, RETURN)
C
INORM=0
IF(NASC=EQ.-2) INORM=1
S1=0
IF(NASC=GT.0) GOTO 20
C
C INITIALIZE FOR NEW RAY
C
DIST=0
IF(KLOOP=LT.32000) GOTO 15
KLOOP=0
LION=LRO+NBODY+NRPP=1
DO 10 I=10+LION
MASTER(I)=0
10 CONTINUE
15 KLOOP=KLOOP+1
C
C BEGIN/CONTINUE TRACING RAY THRU REGION
C
20 SM=PINF
WHIT=0

FIG. 83. Source Listing, Subroutine GI
COMPUTE LOCATION OF REGION DATA
LOC=LREGD+IR-1
RETRIEVE THE NUMBER OF BODIES IN REGION
CALL UN2(LOC,LOC+NC)
LOC=LOC+1
DO 500 N=1,NC
RETRIEVE BODY NUMBER
LOC=LOC+1
CALL UN2(LOC,DUM+NBO)
RETRIEVE ENTER AND EXIT SURFACE NUMBERS AND LAST RAY NUMBER
ITEMP=LIO+NBO-1
CALL UN3(ITEMP,LRI,LRO,LOOP)
RETRIEVE BODY TYPE AND LOCATION OF DATA
ITEMP=LBODY+3*(NBO-1)
CALL UN2(ITEMP,ITY,LOCDA)
IF(LOOP,NE,KLOOP)GOTO 130
CONTINUE RAY RETRIEVE RIN/ROUT FOR CURRENT BODY
IF(ITYPE,GT,11)GOTO 140
IJK=LRIN+NBO-1
RIN=ASTER(IJK)
IJK=LRIN+NBO-1
ROUT=ASTER(IJK)
IF(ITYPE,LT,10)GOTO 320
IS NEXT RIN/ROUT SET REQUIRED FOR TOR OR ARS
IF(ROUT,LT,0,)GOTO 320
IF(NIST,LT,ROUT)GOTO 320
IF(NASC,EQ,NBO)NASC=0
130 LRI=1
LRO=1
ITY=ITYPE+1
IF(ITY,GT,12)GOTO 200
ERR=ERR+1
WRITE (*,901)ITYPE
GOTO 800
COMPUTE RIN/ROUT FOR CURRENT BODY
CALL RPP BOX SPH RCC REC TRC ELL RAW ARB TEC TOR ARS
205 CALL RPP(NBO)
GOTO 300
210 CALL BOX
GOTO 300
215 CALL SPH
GOTO 300
FIG. 83. (Contd.)
220 CALL RCC
GOTO 300
225 CALL REC
GOTO 300
230 CALL TRC
GOTO 300
235 CALL ELL
GOTO 300
240 CALL RAW
GOTO 300
245 CALL ARB
GOTO 300
250 CALL TEC
GOTO 300
255 CALL TOR
GOTO 300
260 CALL ARS

C11 STORE RIN AND ROUT FOR BODY IN RIN AND ROUT TABLES

C12 IS POINT XP ON CURRENT BODY? YES=IS IT ENTER OR EXIT

C13 DOES RAY INTERSECT BODY? YES=DOES IT ORIGINATE WITHIN BODY

C14 POINT XP AT RIN OR WITHIN BODY

C15 POINT XP AT ROUT OF BODY

FIG. 83. (Contd.)
351 IF(RIN=SM)352,355,500
352 IF(DIST<GE.RIN)GOTO 340
   NHIT=0
355 NHIT=NHIT+1
   SM=RIN
   LSURT(NHIT)=LRI
   NASCT(NHIT)=NBO
500 CONTINUE
C
C16 IF(SM.LT.PINF)GOTO 530
C
C17 ERROR=NO INTERSECT
C
   WRITE (6,902)IR
   WRITE (6,903)X8,W8,KLOOP,NBO,LRI,LRO,NHIT,LOOP
   GOTO 700
C
C18 COMPUTE NEW COORDINATES OF POINT XP AND REVISE DISTANCE TRAVELED
C
530 SI=SI+SM-DIST
   DIST=SM
   XP(1)=X8(1)+SM*WB(1)
   XP(2)=X8(2)+SM*WB(2)
   XP(3)=X8(3)+SM*WB(3)
C
C19 IF(NASC.EQ.-2)RETURN
C
C20 DETERMINE REGION THAT POINT XP NOW IN
C
   DO 660 NN=1,NHIT
      NASC=NASCT(NN)
      LSURT=LSURT(NN)
      LTRUE=0
C
C21 COMPUTE LOCATION OF INTERSECTED BODY DATA
C
      LBC=BODY+3*(NASC-1)
      LOC=LOC+1
C
C22 RETRIEVE LOCATIONS OF REGION ENTER/LEAVE TABLE FOR BODY
C
      CALL UN2(LBC,LENT,LEAV)
      LOC=LOC+1
C
C23 RETRIEVE NUMBER OF REGIONS IN ENTRY LIST AND EXIT LIST
C
      CALL UN2(LOC,NENT,NEAV)
C
C24 COMPUTE THE BEGIN AND END OF LIST
C
      IF(LSURF.LE.0)GOTO 600
      J1=LENT
      J2=LENT+NENT-1
      GOTO 610
600 J1=LEAV
      J2=LEAV+NEAV-1
C
C25 ANY REGIONS IN LIST OR IS RAY LEAVING RPP
C
610 IRPRIM=MASTER(J2)

FIG. 83. (Contd.)
IF(J1.LE.J2)GOTO 620
IF(NASC.GT.NRPP)GOTO 700
IF(LSURF)630*700*700
C
Determine Region Point XP Now Entering
C
620 DO 625 J=J1,J2
IRPRIM=MASTER(J)
CALL #O#1(IRPRIM,LSURF,NASC,LTRUE)
IF(LTRUE.GT.0)GOTO 650
CONTINUE
C
Ray Leaving RPP
C
630 CALL RPP2(LSURF,XP,IRP)
IF(IRP.GT.0)GOTO 631
IRPRIM=0
RETURN
C
Retrieve Location/Number of Region Enter List
C
Compute Beginning and End of List
C
631 LTRUE=0
LOC=LBODY+3*IRP+1
LOC=LOC+1
CALL UN2(LOC,LENT,LEAV)
LOC=LOC+1
CALL UN2(LOC,NEXT,NEAV)
J1=LENT
J2=LENT+NEAV-1
IF(J1.GT.J2)GOTO 100
C
Determine Region Point XP Now Entering in New RPP
C
632 DO 632 J=J1,J2
IRPRIM=MASTER(J)
CALL #O#1(IRPRIM,LSURF,IRP,LTRUE)
IF(LTRUE.GT.0)GOTO 650
CONTINUE
640 CALL UN2(LOC,LENT,LEAV)
LOC=LOC+1
CONTINUE
GOTO 700
C
Region Point XP Entering Has Been Determined
C
650 IF(IR.EQ.IPRIM)GOTO 660
IF(S1.EQ.0.)GOTO 660
IF(S1.LT.0.)GOTO 700
IF(ABS(S1).LE.1.0E-6)GOTO 660
IF(ITESTE.EQ.IYES)RETURN
IF(ITESTE.EQ.IYES)RETURN
C
Retrieve Space and Component Code of Region
C
LOC=IRFO+IR-1
CALL UN2(LOC,ICODE,IDENT)
LOC=IRFO+IRPRIM-1
CALL UN2(LOC,ICODE,IDENT)
GOTO 655
C
FIG. 83. (Contd.)
IF(IDENT.EQ.IDENT1)GOTO 660
RETURN
655 IF(ICODE.NE.ICODE1)RETURN
660 I=IPRIM
GOTO 20
C
C30 START OF ERROR DIAGNOSTIC SECTION
C
700 IERR=IERR+1
WRITE (6,904) IERR
C
C31 COMPUTE GRID CELL NUMBER IF G1 NOT CALLED BY VOLUME OR TEST
C
IF(VOLUME.EQ.IYES.OR.TESTG.EQ.IYES)GOTO 705
I=ABS(M/CELSIZ )+.5
IF(H.LT.0.) IH=-IH
IV=ABS(V/CELSIZ )+.5
IF(V.LT.0.) IV=-IV
WRITE (6,905) IH,IV
705 WRITE (6,906) J1,J2,LSURF,NASC,IR,SM,S1,WB,XB
C
C32 COMPUTE COORDINATES OF XP AT TIME OF ERROR
C
XBD1=X8(1)-DIST
XBD2=X8(2)-DIST
XBD3=X8(3)-DIST
WRITE (6,907)XBD,DIST
WRITE (6,908)
NN=NBD0+NRPP
C
C33 PRINT OUT PERTINENT DATA FOR ALL BODIES IN REGION INTERSECTED
C
BY RAY FOR ERROR ANALYSIS
C
DO 750 I=1,NN
LOC/L0+I-1
CALL UNJ/LOC+I1+I2+I3
IF(KLOOP.EQ.I13)GOTO 750
IKJ=LRO/1+1
IR=ASTER(IJK)
IKJ=LRO/1+1
ROUT=ASTER(IJK)
IF(RIN.NE.ROUT)GOTO 710
WRITE (6,910)RIN,I
GOTO 750
C
710 IF(ABS(RIN).NE.PINF)GOTO 720
IF(ABS(ROUT)=PINF)740*750*740
720 IF(RIN=DIST) 730*744*745
730 IF(ROUT=DIST) 741*742*743
C
740 WRITE (6,911)RIN,ROUT/I1+I2+I
GOTO 750
741 WRITE (6,912)RIN,ROUT/I1+I2+I
GOTO 750
742 WRITE (6,913)RIN,ROUT/I1+I2+I
GOTO 750
743 WRITE (6,914)RIN,ROUT/I1+I2+I
GOTO 750
744 WRITE (6,915)RIN,ROUT/I1+I2+I
GOTO 750
FIG. 83. (Contd.)
FIG. 83. (Concluded)
SUBROUTINE WOWI(JREG, LSBUR, NEX, LTRUE)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PLACE/NB(3), NB(3), IR
COMMON/GENM/RBASE,RIN,ROUG,LR,LP,LOR,M,R,P,CELL,DIR,DIR,DIR,DIR
COMMON/NUM/CURRENT,H,ABBR,ABBR,ABBR,NUM1,NUM1,NUM1,NUM1
1 DATA[L1,L2,L3,L4,L5,L6,L7,L8,L9,L10]/NUMBER OR BODY
COMMON/CHAM/GRID1, GRID2, SURF, LN1, LN2, LN3, LN4, LN5, LN6, LN7, LN8, LN9, LN10
CALL UN2(LOC1,LOC2,NC)
RETRIEVE FIRST OPERATOR/BODY FROM LIST
CALL UN2(LOC1,LOC2,NC)
N=1
IOPER=100
RETRIEVE ENTER AND EXIT SURFACE NUMBERS AND NUMBER OF LAST RAY
ITEMP=L10+NBO=1
CALL UN3(ITEMP,LRI1,LR,LOOP)
RETRIEVE BODY TYPE AND LOCATION OF DATA
ITEMP=LBOY+3*(NBO=1)
CALL UN2(ITEMP,ITYPE,LOCDA)
IF(LOOP,NE,KLOOP)GOTO 30
IF(ITYPE,GE,11)GOTO 40
RETRIEVE RIN AND ROUT FOR CURRENT BODY
IJK=LRAIN+NBO=1
RIN=ASTER(IJK)
IJK=LROT+NBO=1
ROUT=ASTER(IJK)
IF(ITYPE,L1,10)GOTO 310
IS NEXT RIN/ROUT SET REQUIRED FOR TOR OR ARS
IF(ROUT,L1,0)GOTO 400
IF(DIST,LE,ROUT)GOTO 310
LRI=1
LRO=1
ITYP=ITYPE
IF(ITYP,GE,1)AND(ITYP,LE,12)GOTO 100
IER=IERRR
WRITE(6,901)ITYPE
RETURN

FIG. 84. Source Listing, Subroutine WOWI
C7 COMPUTE RIN/ROUT FOR CURRENT BODY

C RPP BOX SPH RCC REC TRC ELL RAN ARB TEC TOR ARS
100 GOTO(110*120*130*140*150*160*170*180*190*200*210*220)*ITY
110 CALL RPP(NBO)
   GOTO 300
120 CALL BOX
   GOTO 300
130 CALL SPH
   GOTO 300
140 CALL RCC
   GOTO 300
150 CALL REC
   GOTO 300
160 CALL TRC
   GOTO 300
170 CALL ELL
   GOTO 300
180 CALL RAW
   GOTO 300
190 CALL ARB
   GOTO 300
200 CALL TEC
   GOTO 300
210 CALL TOR
   GOTO 300
220 CALL ARS
C
300 IJK=IJK+NBO-1
   MASTER(IJK)=KLOOP*15*(LRO*6*LRI)
C CR DETERMINE CORRECT RIN/ROUT AND STORE IN ASTERN ARRAY
C
310 IF(ROUT.LE.0.0) GOTO 330
   IF(ABS(RIN-DIST).GT.DIST*1.0E-6) GOTO 320
   RIN=DIST
   GOTO 330
C
320 IF(ABS(ROUT-DIST).LE.DIST*1.0E-6) ROUT=DIST
C
330 IJK=LRIN+NBO-1
   ASTER(IJK)=RIN
   IJK=LROT+NBO-1
   ASTER(IJK)=ROUT
C9 TEST CONDITIONS FOR POINT XB IN REGION UNDER TEST
C
400 IF(IOPER.GT.4) GOTO 500
C C10 (+) OPERATOR TEST RIN.LE.DIST.LT.ROUT POINT XB IN BODY
   IF(RIN.GT.DIST) GOTO 700
   IF(DIST=ROUT).GT.700 GOTO 700
C C11 (-) OPERATOR TEST ROUT.LE.0 OR DIST.LT.RIN OR DIST.GE.ROUT
   POINT XB OUTSIDE OF BODY
   IF(ROUT.LE.0) GOTO 600
   IF(DIST.LT.RIN) GOTO 600
   IF(DIST.EQ.RIN) GOTO 700

FIG. 84. (Contd.)
IF(DIST.LT.ROUT)GOTO 700
C12 CHECK NEXT BODY IN OPERATOR/BODY LIST
C 600 IF(N.GE.NC)GOTO 800
   N=N+1
   LOC=LOC+1
   CALL UN2(LOC,IOPER,NAO)
   IF(IOPER.EQ.1.OR.IOPER.EQ.5)GOTO 800
   GOTO 10
C C13 (OR) OPERATOR TEST
C ALL (+) OR (-) IN (OR) SERIES MUST BE VALID
C 700 IF(IOPER.NE.1.AND.IOPER.NE.5) RETURN
   IF(N.GE.NC) RETURN
   N=N+1
   DO 710 NN=N+NC
   LOC=LOC+1
   CALL UN2(LOC,IOPER,NHO)
   IF(IOPER.EQ.1.OR.IOPER.EQ.5)GOTO 720
   CONTINUE
   RETURN
   720 N=NN
   GOTO 10
C C14 POINT XB WITHIN CURRENT REGION, LTRUE = 1
C 800 LTRUE=LTRUE+1
   RETURN
   END

FIG. 84. (Concluded)
SUBROUTINE RPP(NBO)
DIMENSION PR(6), LR(6), XS(6), LST(6)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PAREM/XB(3), WB(3), IR
COMMON/GEOM/LBASE+RIN+ ROUT+ LRI+ LRO+ PINF+ IERR+ DIST
COMMON/UNCORE/LNRF+ NTRIP+ NSCAL+ NBD+ NRMA+ LTRIP+ LSCL+ LREGO
1 DATA+ LRI+ LROT+ L10+ LOCDA+ L15+ L30+ LB0+ NASC+ KLOOP
EQUIVALENCE (MASTER+ASTER)

C 901 FORMAT(1H0) ERROR IN RPP/L =I10+5X+4MNBO=I10+5X+3H1R=I10/4H XB=3E20+10/4H WB=3E20+10/4H PR=6E20+10/4H LR=6I10)

C SET UP SIX MEMBER ARRAY TO REPRESENT COORDINATE PAIRS

C LST(1)=1
LST(2)=1
LST(3)=2
LST(4)=2
LST(5)=3
LST(6)=3
L=0
PR(1)=0.
PR(2)=0.

C RETRIEVE THE SIX BOUNDARIES OF THE RPP

C DO 10 I=1,6
XS(I)=S(NBO+I)
10 CONTINUE

C DO 100 I=1,6
II=LST(I)
TEMP=XS(I)=XB(II)
IF(WB(II)) 20,100,30
20 IF(TEMP) 40,100,100
30 IF(TEMP LE 0) GOTO 100
40 TRY=TEMP/WB(II)
DO 60 J=1,3
IF(J LQ I) GOTO 60
60 CONTINUE

C COMPUTE INTERSECT/PLANE COORDINATE

C XRY=XB(J)+TRY*WB(J)

C DETERMINE IF INTERSECT OCCURS WITHIN BOUNDARY OF PLANE

C IF((XS(2*J-1)=XRY)*(XRY=XS(2*J))*LT+0.) GOTO 100
60 CONTINUE
L+L=1

C COMPUTE DISTANCE TO INTERSECT POINT

C PR(L)=TRY
LR(L)=I
IF(L EQ 2) GOTO 130
IF(L LQ 2) GOTO 100
WRITE (6901) L+NBO+IR+XB+WB+PR+LR
ROUT=PINF
RETURN
100 CONTINUE

FIG. 85. Source Listing, Subroutine RPP
GOTO 160

C 130 IF(ABS(PP(1)-PR(2)),LE,PR(1),0•0E=6)GOTO 200
   IF(PR(1)-PR(2))140•180•150

C6 COMPUTE RIN, ROUT, AND SURFACE NUMBERS OF INTERSECTS

C 140 RIN=PR(1)
   LRI=LR(1)
   ROUT=PR(2)
   LRO=LR(2)
   RETURN

150 RIN=PR(2)
   LRI=LR(2)
   ROUT=PR(1)
   LRO=LR(1)
   RETURN

C 160 IF(L•0E=1)GOTO 180

C7 ASSIGN VALUE TO ROUT FOR NO INTERSECTION

C 170 ROUT=PINF
   RETURN

C8 RAY ORIGINATES WITHIN RPP

C 180 RIN=PINF
   LRI=0
   ROUT=PR(1)
   LRO=LR(1)
   RETURN

C9 DETERMINE IF RAY ORIGINATES WITHIN RPP OR MISSES

C 200 DO 220 J=1,3
   IF(XB(J)*LT.XS(2*J-1))GOTO 170
   IF(XB(J)*GT.XS(2*J))GOTO 170
   CONTINUE
   GOTO 180
   END

FIG. 85. (Concluded)
SUBROUTINE BOX
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PARENT/XB(3)*WB(3)*IR
COMMON/GEOM/LBASE/RIN*ROUT,LRI,LRO,PINF,IERR,DIST
COMMON/UNCER/NRPP*NTRIP*NSCAL*NBODY*NRMAX*LTRIP*LSCAL*LREGD+
1 DATA,LRI,LROT,LIO,LOCDA*115*I130*L8ODY*NASA*KLOOP
EQUIVALENCE (ASTER,ASTER)

C C C
C RETRIEVE LOCATION OF BOX VERTEX AND M1 COORDINATES
C CALL UNZLOCOCAL,IV,IM1)
LOC=LOCDA+1

C C C
C RETRIEVE LOCATION OF BOX M2 AND M3 COORDINATES
C CALL UNZLOCOCAL,IM2,IM3)
RIN=PINF
ROUT=PINF
ON 105 IM=1+3
IF(I-2)11*12*13
11 IF=2
GOTO 14
12 (IE)
GOTO 14
13 IF=3
14 AM=V
VP=W

C C C
C COMPUTE VECTOR DOT PRODUCTS
C ON 15 J=1+3
JV=IV+J
JW=IW+J
VP=VP+(ASTER(JV-1)*XB(J)))*ASTER(JA-1)
VP=VP+(JW)*ASTER(JA-1)
W=W+(ASTER(JA-1)**2
15 CONTINUE
IF(W)30*20*40
20 IF(W)30*20*40
IF(1-VP*LTS)GOTO 200
IF(VP=A100*I100*200

C C C
C COMPUTE ROUT
C 30 CP=VP/W
LW=2*II-1
IF(CP>LE+0.)GOTO 200

C C C
C COMPUTE RIN
C CM=(VP+V)/W
L=LO+1
GOTO 60

C C C
C COMPUTE ROUT
C 40 CP=(VP+V)/W
LW=2*II
IF(CP>LE+0.)GOTO 200

FIG. 86. Source Listing, Subroutine BOX
COMPUTE RIN

CM=VP/W
LI=LO-1
60 IF(ROUT.LE.CP) GOTO 80
ROUT=CP
LA0=LO
90 IF(RIN.GE.CM) GOTO 100
RIN=CM
LRI=LI
100 IH1=IH2
IH2=IH3
105 CONTINUE
   IF(A:=(RIN= ROUT).LE. ROUT*1.0E-6) GOTO 200
   IF(RIN.LT. ROUT) RETURN

200 RIN=PINF
   ROUT=PINF
   RETURN
   END

FIG. 86. (Concluded)
SUBROUTINE SPH
COMMON ASTER(10000)
COMMON/PAREM/XB(3),WB(3),IR
COMMON/GEOM/LBASE,LR,LRD,PINF,IER=DIR
COMMON/UNCGE/NR,TRIP,NSCAL,NBODY,NMILA,LRTRIP,LSCAL,LREGD
1 LOCA=LR=LRD=LIO=LOCDA=I15=I30=LBODY=NASC,KLOP
C
C RETRIEVE LOCATION OF SPH VERTEX AND RADIUS
C
CALL UN2(QOCO,ITEMP,12)
R=ASTER(I2)
ITEMP=ITEMP+1
DX=XB(1)-ASTER(ITEMP)
DY=XB(2)-ASTER(ITEMP)
DZ=XB(3)-ASTER(ITEMP)
B=DX*WB(1)+DY*WB(2)+DZ*WB(3)
C=DX*DX+DY*DY+DZ*DZ=R*R
DIS=B*B+C
IF(C.GT.0.)GOTO 10
C
C RAY ORIGINATES WITHIN SPHERE

RIN=PINF
ROUT=SQRT(DIS)-R
RETURN

10 IF(DIS.GT.0.)GOTO 20

C
C RAY MISSES SPHERE

RIN=PINF
ROUT=PINF
RETURN

C
C RAY INTERSECTS SPHERE

20 DIS=SQRT(DIS)
RIN=B-DIS
ROUT=B+DIS
RETURN
END

FIG. 87. Source Listing, Subroutine SPH
SUBROUTINE RCC
DIMENSION V(3), H(3)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PAREN/XB(3), WB(3)*IR
COMMON/GEOM/LBASE, RIN, ROUT, LRI, LRO, PINF, IERR, DIST
COMMON/UNCGEM/NAPP, NTRIP, NSCAL, L BODY, NRMAX, LTRIP, LSCAL, LREGO,
1 LDATA, LROUT, LROT, LID0, LOCDA+1, 150+130, LBODY, NASC, KLOOP
EQUIVALENCE (ASTER, MASTER)

C1 RETRIEVE LOCATION OF RCC VERTEX AND HEIGHT VECTOR COORDINATES
CALL UN2(LOCDA+IV, IH)
C2 RETRIEVE LOCATION OF RADIUS
IRR MASTER(LOCDA+1)
C3 RETRIEVE COORDINATES OF VERTEX AND HEIGHT VECTOR
H(1) = ASTER(IH)
H(2) = ASTER(IH+1)
H(3) = ASTER(IH+2)
V(1) = ASTER(IV)
V(2) = ASTER(IV+1)
V(3) = ASTER(IV+2)
C4 RETRIEVE RADIUS
R = ASTER(I RR)
RIN = PINF
ROUT = PINF
C5 COMPUTE R SQUARED
RSQ = R*R
LRO = 0
LRI = 0
TOP = 0.
PRT = 0.
C6 COMPUTE VECTOR DOT PRODUCTS
HH = H(1)*H(1) + H(2)*H(2) + H(3)*H(3)
VPH = V(1)*V(1) + X8(1)*X8(1) + V(2)*V(2) + X8(2)*X8(2) + V(3)*V(3) + X8(3)*X8(3)
WH = WB(1)*H(1) + WB(2)*H(2) + WB(3)*H(3)
C7 COMPUTE COEFFICIENT OF S SQUARED
DEN = HH*WH*WH
DO 10 I=1, 3
TOP = TOP + WB(I)*X8(I) - V(I))
PRT = POT + (X8(I) - V(I))**2
10 CONTINUE
AMD = HH*TOP + WH*VPH
UN = (POT + RSQ)*HH*VPH**2
IF (WH*40 + 70 + 50)
C8 SOLVE FOR RIN AND ROUT OF PLANE INTERSECTIONS
CP = VPH/WH

FIG. 88. Source Listing, Subroutine RCC
SOLVE FOR RIN AND ROUT OF QUADRATIC INTERSECTIONS

R1 = AMBDA = SD
R2 = AMBD*SD

100 IF (CM*GT*R1) GOTO 110
RIN = R1
LRI = 3
GOTO 120

110 RIN = CM
LRI = LC

120 IF (CP*LE*R2) GOTO 130
ROUT = R2
LRO = 3
GOTO 200

130 ROUT = CP
LRO = LP

200 IF (ABS(ROUT-RIN) - LE, ROUT1 + 0, GE, 6) GOTO 300
GOTO (210 + 210 + 220) + LRO

DETERMINE IF ROUT INTERSECTS PLANE WITHIN CYLINDER CROSS-SECTION

210 F1*DEN*ROUT*LE = 2.2 + AMBD = ROUT*UM
IF (F1) 250 + 250 + 300

DOES ROUT INTERSECT OF QUADRATIC SURFACE OCCUR BETWEEN PLANES

220 F1*ROUT*WH = VPH
IF (F1) 300 + 250 + 230
230 IF (F1*GT*HH) GOTO 300
250 GOTO (260 + 260 + 270) + LRI

DETERMINE IF RIN INTERSECTS PLANE WITHIN CYLINDER CROSS-SECTION

260 F1*DEN*RIN*LE = 2.2 + AMBD = RIN + UM
IF (F1) 310 + 310 + 300

FIG. 88. (Contd.)
C13  DOES RIN INTERSECT OF QUADRATIC SURFACE OCCUR BETWEEN PLANES
C
270  F1=RIN=MH=VPH
    IF(F1)300*310*280
280  IF(F1.LE.MH)GOTO 310
C
C14  RAY MISSES BODY
C
300  RIN=PINF
    ROUT=PINF
    LRO=0
    LRI=0
310  RETURN
END
C

FIG. 88. (Concluded)
SUBROUTINE REC

DIMENSION V(3),H(3),A(3),B(3)
COMMON ASTER(10000)
COMMON/REEM/XB(3),WB(3),IR
COMMON/GEOM/LBASE,RIN,ROUR,LRO,PINF,IERD,DIST
COMMON/UNGEM/NRPP,TRAPNSCAL,NL_REFERENCE,LR_MAX,LTRIP,LSCAL,LREGD,
1 LAATA,LRIN,LROT,LLO,LOCDA+L15+L20,LBODY+LASC+KLOOP

C
C RETRIEVE LOCATION OF REC VERTEX AND HEIGHT VECTOR COORDINATES

CALL UN2LocDA+IV, IH)
LOC=LOCDA+1
C
C RETRIEVE LOCATION OF REC COORDINATES FOR AXES

CALL UN2LOC,IA,IB)
C
C RETRIEVE COORDINATES OF VERTEX, HEIGHT VECTOR, SEMI-MAJOR AXIS
AND SEMI-MINOR AXIS

V(1)=ASTER(IV)
V(2)=ASTER(IV+1)
V(3)=ASTER(IV+2)
H(1)=ASTER(IH)
H(2)=ASTER(IH+1)
H(3)=ASTER(IH+2)
A(1)=ASTER(IA)
A(2)=ASTER(IA+1)
A(3)=ASTER(IA+2)
B(1)=ASTER(IB)
B(2)=ASTER(IB+1)
B(3)=ASTER(IB+2)
RIN=PINF
ROUT=PINF
LRO=0
LRI=0

C
C COMPUTE DOT PRODUCTS OF AxB AND BxB

AA=A(1)*A(1)+A(2)*A(2)+A(3)*A(3)
BB=B(1)*B(1)+B(2)*B(2)+B(3)*B(3)

C
C COMPUTE (VxB) FOR XYZ COORDINATES

V1XBl=V(1)-XB(1)
V2XBl=V(2)-XB(2)
V3XBl=V(3)-XB(3)

C
C TRANSFORM XB(X,Y,Z) TO THE COORDINATES OF THE REC

VPA=V1XBl*A(1)+V2XBl*A(2)+V3XBl*A(3)
VPB=V1XBl*B(1)+V2XBl*B(2)+V3XBl*B(3)

C
C TRANSFORM WB(X,Y,Z) TO THE COORDINATES OF THE REC

WB1=WBl*A(1)+WB2*A(2)+WB3*A(3)
WB2=WBl*B(1)+WB2*B(2)+WB3*B(3)
WB3=WBl*B(1)+WB2*B(2)+WB3*B(3)
WB4=WBl*A(1)+WB2*A(2)+WB3*A(3)
WB5=WBl*B(1)+WB2*B(2)+WB3*B(3)

C
C

FIG. 89. Source Listing, Subroutine REC
COMPUTE THE INTERSECT POINTS ON THE QUADRATIC SURFACE

\[ \text{SD} = \sqrt{\text{DISC}} \]
\[ R_1 = \frac{\text{AMBD}}{\text{SD}} \]
\[ R_2 = \frac{\text{AMBD}}{\text{SD}} \]
\[ \text{GOTO 20} \]

10 \[ R_1 = \text{PINF} \]
\[ R_2 = \text{PINF} \]

20 \[ \text{WH} = (H_1)^2 + (H_2)^2 + (H_3)^2 \]
\[ \text{WH} = 1 \cdot (H_1)^2 + (H_2)^2 + (H_3)^2 \]
\[ \text{VPH} = V_1 \cdot (H_1)^2 + V_2 \cdot (H_2)^2 + V_3 \cdot (H_3)^2 \]

DETERMINE IF RAY PARALLEL TO PLANAR SURFACES

\[ \text{IF} (\text{WH} \cdot 1 \cdot 0 \cdot 50) \]
\[ \text{IF} (\text{VPH} \cdot 0 \cdot 0 \cdot 0) \text{GOTO 300} \]

COMPUTE THE INTERSECT POINTS ON THE PLANAR SURFACES

\[ \text{CP} = \text{VPH} \cdot \text{WH} \]
\[ \text{CM} = (\text{VPH} \cdot \text{WH}) \cdot \text{WH} \]
\[ \text{LCP} = 1 \]
\[ \text{LCM} = 2 \]
\[ \text{GOTO 100} \]

50 \[ \text{VPH} = \text{WWW} \]
\[ \text{IF} (\text{VPH} \cdot \text{LE} \cdot 0) \text{GOTO 300} \]
\[ \text{CP} = \text{VPH} \cdot \text{WH} \]
\[ \text{CM} = \text{VPH} \cdot \text{WH} \]
\[ \text{LCP} = 1 \]
\[ \text{LCM} = 2 \]
\[ \text{GOTO 100} \]

70 \[ \text{CM} = \text{PINF} \]
\[ \text{CM} = \text{CP} \]
\[ \text{IF} (\text{CM} \cdot \text{GT} \cdot \text{R1}) \text{GOTO 110} \]

RIN FOR THE QUADRATIC SURFACE

\[ \text{RIN} = \text{R1} \]
\[ \text{LRI} = 3 \]
\[ \text{GOTO 120} \]

RIN FOR A PLANAR SURFACE

\[ \text{RIN} = \text{CM} \]
\[ \text{LRI} = \text{LCM} \]
\[ \text{IF} (\text{CP} \cdot \text{LE} \cdot \text{R2}) \text{GOTO 130} \]

ROUT FOR THE QUADRATIC SURFACE

\[ \text{ROUT} = \text{R2} \]

FIG. 89. (contd.)
ROUT FOR A PLANAR SURFACE

130 ROUT=CP
   LRO=LCP
200 IF(ABS(ROUTE-RIN)*LE.ROUTE*1.0E-5)GOTO 300
   GOTO(210+210+220)*LRO

DETERMINE IF ROUT OF PLANAR SURFACE OCCURS WITHIN ELLIPTIC CROSS-SECTION

210 F1=DEN*ROUTE**2-2.*AMBO*ROUTE*UM
   IF(F1)250*250*300

DETERMINE IF ROUT OF QUADRATIC OCCURS BETWEEN PLANAR SURFACES

220 F1=ROUTE*WH=VPH
   IF(F1)300*250*230
230 IF(F1.GT.*HM)GOTO 300
250 GOTO(260+260+270)*LRI

DETERMINE IF RIN OF PLANE WITHIN ELLIPTIC CROSS SECTION

260 F1=DEN*RIN**2-2.*AMBD*RIN*UM
   IF(F1)310*310*300

DETERMINE IF RIN OF QUADRATIC SURFACE BETWEEN PLANAR SURFACES

270 F1=RIN*WH=VPH
   IF(F1)300*310*280
280 IF(F1.LE.*HM)GOTO 310

RAY MISSES BODY

300 RIN=PINF
   ROUT=-PINF
   LRI=0
   LRO=0
310 RETURN

END

FIG. 89. (Concluded)
SUBROUTINE TRC

DIMENSION V(3), H(3)
DIMENSION MASTER(10000)
COMMON MASTER(10000)
COMMON/ASTER/I
COMMON/ASTER/I
COMMON/ASTER/I
COMMON/ASTER/I
COMMON/ASTER/I
COMMON/ASTER/I
COMMON/ASTER/I
COMMON/ASTER/I

C1 RETRIEVE LOCATION OF TRC VERTEX AND HEIGHT VECTOR COORDINATES

CALL UNZ(LOCDA+IV, IH)
LOC = LOCDA+1

C2 RETRIEVE LOCATION OF TRC RADII FOR LOWER BASE AND UPPER BASE

CALL UNZ(LOC+IRB, IRTOP)

C3 RETRIEVE COORDINATES OF VERTEX AND HEIGHT VECTOR

V(1) = MASTER(IV)
V(2) = MASTER(IV+1)
V(3) = MASTER(IV+2)
H(1) = MASTER(1H)
H(2) = MASTER(2H)
H(3) = MASTER(3H)

C4 RETRIEVE RADII OF LOWER AND UPPER BASES

RB = MASTER(IRB)
RT = MASTER(IRTOP)
PINF = PINF
ROUT = PINF
LRI = 0
LRO = 0
INTSEC = 0
INTR1 = 0
INTR2 = 0

C5 COMPUTE COORDINATES OF (V=XB)

V1XB1 = V(1) - XB(1)
V2XB2 = V(2) - XB(2)
V3XB3 = V(3) - XB(3)

C6 COMPUTE DOT PRODUCTS

VPV = V1XB1 * V1XB1 + V2XB2 * V2XB2 + V3XB3 * V3XB3
VPW = V1XB1 * WB(1) + V2XB2 * WB(2) + V3XB3 * WB(3)
WH = WB(1) * H(1) + WB(2) * H(2) + WB(3) * H(3)

C7 COMPUTE C2 QUANTITY OF QUADRATIC EQUATION

RBRTVP = RB * VPN * RT / WH
VPNMM = VPN * MM
UMMM = UM * (VPV = RBRTVP * 2) = VPN * VM
AMBD = MM * VPN = MM * (VPN * RBRTVP)

FIG. 90. Source Listing, Subroutine TRC
DEN=HH=WW@2*(1+RTR8@2/HH)

TEST FOR RAY PARALLEL TO EITHER SIDE OF CONE
IF(ABS(DEN).GT.1.0E-6)GOTO 40
IF(RTR8.EQ.0)GOTO 200

COMPUTE INTERSECT WITH QUADRATIC SURFACE FOR RAY PARALLEL TO SIDE
R2=UM/(2*AMBD)
F1=R2*WH=VPH

TEST IF INTERSECT BETWEEN PLANAR SURFACES
IF(F1.LT.0.)GOTO 200
IF(F1.GT.MH)GOTO 200
INTSEC=INTSEC+1
IF(WH.LE.0.)GOTO 10
IF(RTR8)=20*20+30
10 IF(RTR8)=30*30+20

ASSIGN SURFACE EXIT NUMBER AND ROUT FOR QUADRATIC SURFACE
20 LRO=3
ROUT=R2
GOTO 250

ASSIGN SURFACE ENTRY NUMBER AND RIN FOR QUADRATIC SURFACE
30 LAI=3
RIN=R2
INTSEC=INTSEC+1
GOTO 210

AMBD=AMBD/DEN
UMU=UM/DEN
DISC=AMBD**2-UMU
IF(DISL)=350*200+50

SOLVE FOR VALUES OF QUADRATIC EQUATION
50 SD=SQRT(DISL)
R1=AMBD=SD
R2=AMBD=SD
F1=R2*WH=VPH

TEST FOR INTERSECT BETWEEN PLANAR SURFACES
IF(F1.LT.0.)GOTO 60
IF(F1.LE.HH)INTR2=INTR2+1
60 F1=R1*WH=VPH
IF(F1.LT.0.)GOTO 70
IF(F1.LE.HH)GOTO 80
70 IF(INTR2.LE.LT.1)GOTO 200
ROUT=R2
RIN=R2
LRO=3
LAI=3
INTSEC=INTSEC+1
GOTO 200

FIG. 90. (Contd.)
80 INTR1=INTR1+1
   IF (INTR2 GE 1) GOTO 90
   ROUT=R1
   RIN=R1
   LRA=3
   LAI=3
   INTSEC=INTSEC+1
   GOTO 200
90 IF (R1=R2) 100, 350, 110

C 15 COMPUTE RIN AND ROUT FOR QUADRATIC SURFACE

C 100 RIN=R1
   ROUT=R2
   LRA=3
   LAI=3
   GOTO 300
110 RIN=R2
   ROUT=R1
   LRA=3
   LAI=3
   GOTO 300

C 200 IF (WH) 210, 350, 250
210 IF (VPH GE 0.1) GOTO 350
   CP=VPH/WH
   F1=CP*CP-2.*CP*VPV+VPV=RB*RB
   IF (F1 GT 0.) GOTO 220

C 16 COMPUTE ROUT FOR EXIT FROM V=PLANE SURFACE

C INTSEC=INTSEC+1
   ROUT=CP
   LRA=1
   IF (INTSEC GE 2) GOTO 300
220 CM=VPH/MM/WH
   F1=CM*CM-2.*((VPW+WH)*CM=VPH)+MM*VPV=RT*RT
   IF (F1 LT 0.) GOTO 260

C 17 COMPUTE RIN FOR ENTRY INTO V=H PLANE SURFACE

C RIN=CM
   LRA=2
   GOTO 300
250 IF (VPHMM LT 0.) GOTO 350
   CP=VPHMM/WH
   F1=CP*CP-2.*((VPW+WH)*CP=VPH)+MM*VPV=RT*RT
   IF (F1 GT 0.) GOTO 260

C 18 COMPUTE ROUT FOR EXIT FROM V=H PLANE SURFACE

C INTSEC=INTSEC+1
   ROUT=CP
   LRA=2
260 IF (INTSEC GE 2) GOTO 300
   CM=VPH/WH
   F1=CM*CM-2.*CM*VPW+VPV=RB*RB
   IF (F1 LT 0.) GOTO 350

FIG. 90. (Cont.)
COMPUTE RIN FOR ENTRY INTO V-PLANE SURFACE

RIN=CM
LRI=1

300 IF(ABS(ROUT-RIN)-ROUT*1.E5)350,350,360

RAY MISSES TRC

350 RIN=PINF
ROUT=PINF
LRI=0
LRO=0
360 RETURN
END

FIG. 90. (Concluded)
SUBROUTINE ELL
DIMENSION FOCIA(3),FOCIB(3)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PAREM/XB(3),WB(3),IR
COMMON/GEOM/LBASE,LR1,LR0,PINF,IERR,DIST
COMMON/UNCinem/NRPP,NTRIP,NSCAL,NBODY,NRMAX,LTRIP,LSCAL,LR60
1, LDATA,LRIN,LROT,L10,LOCDA+115+130,LBODY+100+KLOOP
EQUIVALENCE (ASTER+MASTER)

RETREIVE LOCATION OF ELLIPSE FOCI AND LENGTH STORAGE POSITIONS

CALL UN2(LOCDA+IV1+IV2)
IRR=MASTER(LOCDA+1)
FOCIA(1)=ASTER(IV1)
FOCIB(1)=ASTER(IV2)
FOCIA(2)=ASTER(IV1+1)
FOCIB(2)=ASTER(IV2+1)
FOCIA(3)=ASTER(IV1+2)
FOCIB(3)=ASTER(IV2+2)
C=ASTER(IRR)
RIN=PINF
ROUT=PINF

COMPUTE COORDINATES FOR VECTOR D1 AND D2

D1X=XB(1)-FOCIA(1)
D1Y=XB(2)-FOCIA(2)
D1Z=XB(3)-FOCIA(3)
D2X=XB(1)-FOCIB(1)
D2Y=XB(2)-FOCIB(2)
D2Z=XB(3)-FOCIB(3)

COMPUTE DOT PRODUCTS

A1=(D1X*W8(1)+D1Y*W8(2)+D1Z*W8(3))
A2=(D2X*W8(1)+D2Y*W8(2)+D2Z*W8(3))
B1=D1X*W1X+D1Y*W1Y+D1Z*W1Z
B2=D2X*W1X+D2Y*W1Y+D2Z*W1Z

COMPUTE A AND B

AA=(A2-A1)/(2.*C)
BB=(C*CC*B2-B1)/(2.*C)

COMPUTE LAMBOA AND MU

ALAMD=AA+AA=1.
ALAM1=(AA*BB-5*A2)/ALAMD
U=(BB*BB-B2)/ALAMD

COMPUTE RIN AND ROUT

DISCRM=ALAM1*ALAM1=U
IF (DISCRM.LE.0.)RETURN
SORTDI=SQRT(DISCRM)
RIN=ALAM1=SORTDI
ROUT=ALAM1=SORTDI
RETURN
END

FIG. 91. Source Listing, Subroutine ELL
SUBROUTINE RAW
DIMENSION H1(3),H2(3),H3(3),V(3),ASQ(3),PV(4),G(3)
COMMON ASTER(10000)
COMMON/PARE/XB(3),WB(3),IR
COMMON/GEOM/LBASE,IRN,RUTL,RROT,LRO,PINF,INR,DIST
COMMON/UNCO/GNP,TRIP,NSCAL,NOEBO,NDM,WMAX,LTRIP,LSCL,LREGD
1 LOATA+LROT+L0+LOCDA=I15+I30+L800+NASC+KLOOP

RETRIEVE LOCATIONS OF VERTEX AND LENGTH VECTORS

CALL UNZ2(LOCDA,IV,IM1)
LOC=LOCDA+1
CALL UNZ2(LOC,II2,II3)
H1(1)=ASTER(IM1)
H1(2)=ASTER(IM1+1)
H1(3)=ASTER(IM1+2)
H2(1)=ASTER(IM2)
H2(2)=ASTER(IM2+1)
H2(3)=ASTER(IM2+2)
H3(1)=ASTER(IM3)
H3(2)=ASTER(IM3+1)
H3(3)=ASTER(IM3+2)
V(1)=ASTER(IV)
V(2)=ASTER(IV+1)
V(3)=ASTER(IV+2)
RIN=PINF
ROUT=PINF
CM=PINF
CF=PINF
L=0
L1=0
K=0
LR1=0
LR0=0

COMPUTE A
ASQ(1)=H1(1)*H1(1)+H1(2)*H1(2)+H1(3)*H1(3)
ASQ(2)=H2(1)*H2(1)+H2(2)*H2(2)+H2(3)*H2(3)
ASQ(3)=H3(1)*H3(1)+H3(2)*H3(2)+H3(3)*H3(3)

COMPUTE P
X81V1=X8(1)=V(1)
X82V2=X8(2)=V(2)
X83V3=X8(3)=V(3)
P13V1=X81V1*H1(1)+X82V2*H1(2)+X83V3*H1(3)
P13V2=X81V1*H2(1)+X82V2*H2(2)+X83V3*H2(3)
P13V3=X81V1*H3(1)+X82V2*H3(2)+X83V3*H3(3)

COMPUTE G
G(1)=WB(1)*H1(1)+WB(2)*H1(2)+WB(3)*H1(3)
G(2)=WB(1)*H2(1)+WB(2)*H2(2)+WB(3)*H2(3)
G(3)=WB(1)*H3(1)+WB(2)*H3(2)+WB(3)*H3(3)

G. 140 I=1+2
10 IF(G(1))10 10 110+60
10 IF(PV(1))10 120+400+400

FIG. 92. Source Listing, Subroutine RAW
C C6 COMPUTE S1 OR S3
C 20 TEMP=PV(I)/G(I)
IF (TEMP=CP) 30 +130 +130
30 CP=TEMP
L=1
GOTO (40 +50) +1
40 LRO=3
GOTO 130
50 LRO=1
GOTO 130
60 IF (=PV(I)+LE.+0.) GOTO 130
C C6 COMPUTE S1 OR S3
C TEMP=PV(I)/G(I)
IF (TEMP=LE.+CM) GOTO 130
CM=TEMP
K=1
GOTO (90 +100) +1
90 LRI=3
GOTO 130
100 LRI=1
GOTO 130
C 110 IF (PV(I)+LE.+0.) GOTO 810
IF (PV(I)+GE.+ASQ(I)) GOTO 810
130 L1=L1+1
140 CONTINUE
C C7 COMPUTE S6
C 150 TEMP=ASQ(3)=PV(3)
IF (TEMP=LE.+0.) GOTO 180
TEMP=TEMP/G(3)
IF (TEMP=LE.+CM) GOTO 190
CM=TEMP
K=3
LRI=6
180 IF (=PV(3)+) 190 +400 +400
C C9 COMPUTE S5
C 190 TEMP=PV(3)/G(3)
IF (TEMP=LE.+CP) GOTO 290
CP=TEMP
L=3
LRO=5
GOTO 290
C 210 IF (PV(3)+LE.+0.) GOTO 400
IF (PV(3)-ASQ(3)) 290 +290 +400
C C9 COMPUTE S5
C 230 IF (=PV(3)+LE.+0.) GOTO 260
TEMP=-PV(3)/G(3)
FIG. 92. (Contd.)
IF (TEMP * LE. * CM) GOTO 260
CM = TEMP
K = 3
LRI = 5

C10 COMPUTE S6

260 TEMP = ASQ (3) * PV (3)
IF (TEMP * LE. * 0.) GOTO 400
TEMP = TEMP / (3)
IF (TEMP * GE. * CP) GOTO 290
CP = TEMP
L = 3
LRI = 6

C11 COMPUTE S2

290 AG = ASQ (2) * G (1) * ASQ (1) * G (2)
Pv (4) = PV (1) * ASQ (2) + PV (2) * ASQ (1)
TOP = ASQ (1) * ASQ (2) * PV (4)
IF (AG) 310 * 350 + 330

310 TEMP = TOP / AG
IF (TEMP * LE. * CM) GOTO 380
CM = TEMP
K = 4
LRI = 2
GOTO 380

C330 IF (TOP * LT. * 0.) GOTO 400
TEMP = TOP / AG
IF (TEMP = CP) 370 * 380 + 380

C350 IF (PV (4) * LE. * 0.) GOTO 400
IF (TOP) 380 * 400 + 400

370 CP = TEMP
L = 4
LRI = 2
380 IF (L * K * LE. * 0.) GOTO 400
ROUT = CP
RIN = CM

C400 IF (ROUT * GE. * PINF) GOTO 810
IF (ROUT * LE. * 0.) GOTO 810
IF (RIN * GE. * ROUT) GOTO 810
IF (ABS (RIN - ROUT) * G T. * ROUT * 1. * OEG) GOTO 820

C810 ROUT = PINF
RIN = PINF
LRI = 0
LRO = 0
820 RETURN
END

FIG. 92. (Concluded)
SUBROUTINE ARB

DIMENSION AA(6,4), XP(3)
COMMON ASTER(10000)
COMMON/PARAM/XB(3), W(3), IR
COMMON/GEOM/LBASE, RIN, ROUT, LRI, LRO, PNF, IERR, DIST
COMMON/UNCORE/NRRP, NTRIP, NSCAL, NBODY, NMAX, LTRIP, LSCALE, LREGO,
1 LOAT, LRRN, LROT, LDD, LOCDA, I15, I30, LMAX, NSCALE, KLOOP

C
C RETRIEVE PLANAR EQUATIONS FROM ASTER ARRAY

LOC = LOCDA = 1
DO 10 I = 1, 6
LOC = LOC + 1
CALL UNZCLOC (LOC, LD, LC)
AA(I+1) = ASTER(LC)
AA(I+2) = ASTER(LC+1)
AA(I+3) = ASTER(LC+2)
AA(I+4) = ASTER(LD)
10 CONTINUE
RIN = PNF
ROUT = PNF
LRO = 0
LRI = 0
S1 = 0
S2 = 0
L1 = 0
L2 = 0
DO 70 I = 1, 6

C

C COMPUTE NUMERATOR AND DENOMINATOR OF DISTANCE EQUATION

D = AA(I+4)
SNUM = D - AA(I+1)*XB(1)*AA(I+3)*XB(3)
SDEN = AA(I+1)*WB(1) + AA(I+2)*WB(2) + AA(I+3)*WB(3)
IF (SDEN) .GT. 0.70 + 30
20 IF (SNUM) .GT. 0.70 + 70
30 IF (SNUM) .GT. 0.70 + 40

C

C COMPUTE INTERSECT DISTANCE

40 S = SNUM / SDEN
DO 50 K = 1, 3
XP(K) = XB(K)*S + WB(K)
50 CONTINUE

C

C TEST IF INTERSECT POINT IS ON ARB

60 DO 60 J = 1, 6
IF (I.EQ.J) GOTO 60
T = AA(J+1)*XP(1) + AA(J+2)*XP(2) + AA(J+3)*XP(3) + AA(J+4)
IF (ABS(T) .LT. 1.0E-6) T = 0
IF (T .GT. 0.70) GOTO 70
60 CONTINUE
IF (L1 .GT. 0.0) GOTO 65
L1 = I
S = 5
GOTO 70
65 IF (ABS(S1 - S) .GT. 1.0E-6) GOTO 100
70 CONTINUE

C

FIG. 93. Source Listing, Subroutine ARB
IF(L1) 200+200+150
100 S2=S
L2=I
IF(ABS(S1)-S2)*LE.S1*1.0E=5) GOTO 200
IF(S1=S2)110+200+120
110 RIN=S1
ROUT=S2
LRI=L1
LRO=L2
RETURN
120 RIN=S2
LRI=L2
130 ROUT=S1
LRO=L1
RETURN
150 DO 160 J=1,6
IF(L1*E=6) GOTO 160
T1=AA(J)+XB(J)+AA(J)+XB(J)+AA(J)+XB(J)
IF(ABS(T1)*LE.1*0E=6) T1=0*
IF(T1+LT.0) GOTO 200
160 CONTINUE
GOTO 130
C
200 RIN=PINF
ROUT=PINF
LRI=0
LRO=0
RETURN
END

FIG. 93. (Concluded)
SUBROUTINE TEC
DIMENSION VXBC(3),MM(3),AA(I3),BB(I3)
DIMENSION MASTER(I0000)
COMMON ASTBTRAN(I0000)
COMMON/GEOM/MBASE+RINT+LRI+LR0+PINF+ERR+DIST
COMMON/NORM/PNRPP+TRIP+NSCAL+NBODY+NRMAX+LTRIP+LSCAL+LRG0:
1 LDATA+LRIN+LRTR+LIO+LOCDA+I1S+I30+LBODY+MA5C+KLOO:
EQUIVALENCE(ASTER,MASTER)

C RETRIEVE LOCATION OF VERTEX AND HEIGHT VECTOR COORDINATES
CALL UN2ILOCDA,IV,1H)
LOC=LOCDA+1

C RETRIEVE LOCATION OF NORMAL AND AXES COORDINATES
CALL UN2ILOC,IN,IA) 
LOC=LOC+1

C RETRIEVE LOCATION OF LENGTHS OF SEI-MAJOR AXIS AND
SEMI-MINOR AXIS OF BASE ELLIPSE
CALL UN2ILOC+I1,1R2)

C RETRIEVE LOCATION OF THE RATIO OF THE LARGER TO SMALLER ELLIPSE
IR3=MASTER(LOC+1)

C RETRIEVE COORDINATES OF VERTEX AND COMPUTE COORDINATES
OF (1-XB) VECTOR
VXBC(I)=ASTER(IV)-XB(I)
VXBC(2)=ASTER(IV+1)-XB(2)
VXBC(3)=ASTER(IV+2)-XB(3)

C RETRIEVE COORDINATES OF HEIGHT VECTOR
M(I)=ASTER(IH)
M(2)=ASTER(IH+1)
M(3)=ASTER(IH+2)

C RETRIEVE COORDINATES OF NORMAL TO BASE ELLIPSE
MN(I)=ASTER(IN)
MN(2)=ASTER(IN+1)
MN(3)=ASTER(IN+2)

C RETRIEVE COORDINATES OF SEMI-MAJOR AXIS OF BASE ELLIPSE
AA(I)=ASTER(IA)
AA(2)=ASTER(IA+1)
AA(3)=ASTER(IA+2)

C COMPUTE SEMI-MINOR AXIS UNIT VECTOR OF BASE ELLIPSE
CALL CROSS(BB,AA,MN)

FIG. 94. Source Listing, Subroutine TEC
C10 REVERSE LENGTHS OF SEMI-MAJOR AND SEMI-MINOR AXES OF BASE ELLIPSE
AND RATIO OF LARGER TO SMALLER ELLIPSE

R1=ASTER(IR1)
R2=ASTER(IR2)
R3=ASTER(IR3)

C11 COMPUTE LENGTHS OF SEMI-MAJOR AND SEMI-MINOR AXES OF TOP ELLIPSE

R3=R1/RR
R4=R2/RR

C12 START OF COMPUTATIONS FOR DOT PRODUCTS

MDN=DOT(H+HN)
MDA=DOT(H+AA)
MDB=DOT(H+BB)
WDN=DOT(WB+HN)
WDA=DOT(WB+AA)
WDB=DOT(WB+BB)
VXBDN=DOT(VXB+HN)
VXBDA=DOT(VXB+AA)
VXBDB=DOT(VXB+BB)

C13 TEST TO DETERMINE IF RAY IS PARALLEL TO TOP AND BASE PLANES

IF(ABS(WDN).GT.0.0001)GOTO 20

C14 COMPUTE RATIO OF NORMAL TO HEIGHT OF HIT

GAMMA = VXBDN/MDN
IF(GAMMA. LT.0.0 OR. GAMMA. GT.1.0) GOTO 500
A=GAMMA*R3/R1*(1.-GAMMA)
B=GAMMA*R4/R2*(1.-GAMMA)
ASQ=A*A
BSQ=B*B
TA=VXBDA*GAMMA*MDA
TB=VXBDN*GAMMA*MDB
DEN=BSQ*MDA+MDA+ASQ*WDB*WDB
IF(ABS(DEN). LE.0.0001) GOTO 500
AMBDA=BSQ*MDA*TA+ASQ*WDB*TB
UM=BSQ*TA+ASQ*TB+TB-ASQ*BSQ
DISC=AMBDA*AMBDA-UM
IF(DIS*LT.0.0) GOTO 500
DISC=SQR(DISC)

C15 COMPUTE RIN AND ROUT AND ASSIGN SURFACE NUMBER FOR RIN AND ROUT WITH QUADRATIC SURFACE

RIN=(AMBDA/DISC)/DEN
ROUT=(AMBDA+DISC)/DEN
LRI=3
LRQ=3
GOTO 400

C16 SOLVE FOR TERMS IN QUADRATIC EQUATION

20 TAU=(R1/R2)**2
R25Q=R2**2

FIG. 94. (Contd.)
TN 4565-3-71 Vol II

TR2SQ=TAU*R2SQ
TR4R2=TAU*(R2-R4)**2
TRR&2=TR2SQ+TAU*R2*R4
BETA=VXBDA/WDN
ALPHA=HDB/WDN
TA1=ALPHA*WDA-HDA
TA1=ALPHA*WDB-HDB
TA2=VXBDA-BETA*WDA
TA2=VXBDA-BETA*WDB
DEN =TA1*TA1+TAU*TB1*TB1+TRR&2
AMBDA=TA1*TA2+TAU*TB1*TB1+TRR&2
UM =TA2*TA2+TAU*TB2*TB2-TR2SQ
IF(ABS(DEN))GE(0.001)GOTO 150
IF(R1.EQ.R3)GOTO 100
IF(AMBDRA=NE+O.0)GOTO 110
C 17 THE RAY MISSES THE QUADRATIC SURFACE OF THE TEC
C 100 S1=PINF
S2=PINF
GOTO 200
110 UM=UM/(2*AMBDA)
C 18 COMPUTE DISTANCE TO INTERSECT WITH QUADRATIC SURFACE
C 120 S=BETA+ALPHA*T
F=S*WDA/VXBDA
IF(ABS(F))LE(0.001)GOTO 125
IF(F*LT.0)GOTO 100
IF(ABS(F))LE(0.001)GOTO 125
IF(F*GT.0)GOTO 100
125 IF(WDN)130=500+140
C 19 ASSIGN TEMPORARY VALUES TO R(IN AND OUT PER DIRECTION OF RAY
C 130 S1=S
S2=PINF
GOTO 200
140 S1=PINF
S2=S
GOTO 200
C 20 RAY PARALLEL TO SIDE
C 150 DISC=AMBDA*AMBDA-DEN*UM
IF(ABS(DISC))GE(0.001)GOTO 155
T=AMBDA/DEN
GOTO 120
155 IF(DISC*LT.0)GOTO 500
C 21 SOLVE FOR TWO INTERSECTS WITH QUADRATIC SURFACE
C DISC=SQR(DISC)
T1=(AMBDA-DISC)/DEN
T2=(AMBDA+DISC)/DEN
S1=BETA+ALPHA*T1
S2=BETA+ALPHA*T2
IF(WDN*GE.0)GOTO 160
T=S1
S1=S2
S2=T

FIG. 94. (Contd.)

684
C22 DETERMINE IF SIDE INTERSECTION BETWEEN PLANES
C
160 F=S1*WDN-VXBON
170 IF(F.LT.0.0)GOTO 170
180 IF(F.LE.WDN)GOTO 180
190 S2=PINF
200 IF(WDN).LT.210+230
C23 RAY PARALLEL TO PLANES
C
210 S1=PINF
220 S0=PINF
230 GOTO 300
C24 COMPUTE INTERSECTIONS WITH PLANE SURFACES
C
220 SI=BETA+ALPHA
230 S0=BETA
240 LT=2
250 LNE=1
260 GOTO 240
C
270 S1=BETA
280 S0=BETA+ALPHA
290 LT=1
300 LNE=2
310 IF(S0.LT.0.0)GOTO 500
C25 DETERMINE WHICH SURFACE IS HIT
C
300 IF(S1.GE.S1)GOTO 310
310 IF(ABS(S1-S1).LE.0.0001)GOTO 310
320 PIN=S1
330 LRI=3
340 GOTO 360
310 RIN=S1
350 LRI=L1
360 IF(S0.LE.S2)GOTO 360
370 IF(ABS(SO-S2).LE.0.0001)GOTO 360
380 ROUT=S2
390 LRO=3
400 GOTO 400
410 ROUT=S0
420 LRO=10
430 IF(RIN.GE.ROUT)GOTO 500
440 IF(ABS(RIN-ROUT).LE.0.0001)GOTO 500
450 GOTO 500
C
460 S=ROUT
470 LRI=1
480 GOTO(420*430*410)*LRO
490 S=RIN
500 LRI=2
510 GOTO(420*430*480)*LRI

FIG. 94. (Contd.)
Determine if intersection with base plane lies within cross section of base ellipse

420 F1=S*WDA-VXDBA
F2=S*WDB-VXDBB
F=F1°F1/(R1°R1)+F2°F2/(R2°R2)
IF(F.GT.1.0001)GOTO 500
GOTO (410+480)+1

430 IF(R3.EQ.0.0.OR.R4.EQ.0.0)GOTO 480

Determine if intersection with top plane lies within cross section of top ellipse

F1=S*WDA-VXDBA-MDA
F2=S*WDB-VXDBB-MDB
F=F1°F1/(R3°R3)+F2°F2/(R4°R4)
IF(F.GT.1.0001)GOTO 500
GOTO (410+480)+1

Ray originates within TEC

480 IF(RIN.GT.0.0001)RETURN
RIN=0.0001
LAI=0
RETURN

Ray misses TEC

500 RIN=PINF
ROUT=PINF
LAI=0
LRO=0
RETURN

End
SUBROUTINE ARS

SUBROUTINE COMPUTES INTERSECTIONS OF RAY WITH ARBITRARY SURFACE - ARS

DIMENSION W(3), VW(3), MXB(3), WN(3),
1 HIT(20), NORMAL(3, 20), ISURF(20)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PAREM/XB(3), WB(3), IR
COMMON/GEOM/MBASE, RIN, ROUT, LRI, LRO, PINF, IERR, DIST
COMMON/UNCRE/ANPP, NTRIP, NSCAL, NBODY, NMAX, LTRIP, MLSCL, LREG,
1 LDATA, LRIN, LROT, LIO, LUCDA, L15, L130, L800, NASG, KLOOP
COMMON/DAVIS/GRID, LOOP, INORM
COMMON/WHICH/NOO
EQUIVALENCE (MASTER, ASTER)

901 FORMAT(1HO, 12HE) ERROR IN ARS, I5, 4X, 22 NUMBER OF HITS, GT, 20
902 FORMAT(1HO, 12HE) ERROR IN ARS, I5, 4X, 21 NUMBER OF HITS IS ODD,
1 2X, 6H(NHIT = I5, 1H)
903 FORMAT(1HO, 12HE) ERROR IN ARS, I5, 4X, 27 WRONG SEQUENCE IN HIT TABLE,
1 2X, 6H(NHIT = I5, 1H)
910 FORMAT(5X, 6H) THIS ERROR USUALLY MEANS THE ARS IS NOT CLOSED /
1 9X, 3H(NHIT, 5X, 7MSURFACE / (F12.4*I12) )

ARS DATA STORAGE IN ASTER ARRAY -

LOCAL

+1 NP = NUMBER OF POINTS
+1 NHIT = NUMBER OF HITS
+2 )
+ ) = 80 WORDS FOR HITS (4 PER HIT)
+ ) = ALLOWS FOR 10 PAIRS OF RIN/ROUT
+ )
LOCHTS = LOCALS + 2
+0 S = DISTANCE FROM START POINT XB TO TRIANGLE HIT
+1 NX = DIRECTION COSINES OF NORMAL TO TRIANGLE HIT
+2 NY = (NX, NY, NZ)
+3 NJ =
+ )
+ ) = 4 WORDS PER POINT (X, Y, Z)
+ ) = NP IS TOTAL NUMBER OF POINTS
+ ) =
+ ) = SET = -1 TO SIGNAL LINE OR POINT TRIANGLE
LOC = LOCALS + 82
+0 X
+1 Y
+2 Z
+3 FLAG
LOC = LOC + 4 FOR NEXT TRIANGLE

DETERMINE IF RE ENTRY

LOCALS = MASTER(LOCDA)
LOCHTS = LOCALS + 2
IF (LOOP .NE. KLOOP) GOTO 100

FIG. 95. Source Listing, Subroutine ARS
REENTRY

NHT=MASTER(LOCARS+1)
IF(NHT.LE.0)RETURN
LOC=LOCHTS

MOVE INTERSECT DATA TO HIT ARRAY

DO 10 I=1,NHT
HIT(I)=ASTER(LOC)
LOC=LOC+4
10 CONTINUE
GOTO 600

NOT A REENTRY - ZERO INTERSECT DATA SECTION OF ASTER ARRAY

100 NHT=0
N=1
IF(NASC.EQ.-2)GOTO 400
L1=LOCHTS
L2=LOCHTS+79
DO 110 L=L1+L2
ASTER(L)=0.
110 CONTINUE

COMPARE FOR HIT ON TRIANGLE IT, STORE FLAG AT ASTER(LOC+1)
-1 TRIANGLE IS A POINT OR LINE (DEGENERATE)
0 NON-DEGENERATE, COMPUTE INTERSECT DATA

400 LOC=LOCARS+82
NT=MASTER(LOCARS)-2
DO 490 IT=1,NT
IF(ASTER(LOC+3)*LT.0,7)GOTO 490
W(1)=ASTER(LOC)
W(2)=ASTER(LOC+1)
W(3)=ASTER(LOC+2)
UW(1)=ASTER(LOC+4)-W(1)
UW(2)=ASTER(LOC+5)-W(2)
UW(3)=ASTER(LOC+6)-W(3)
VW(1)=ASTER(LOC+8)-W(1)
VW(2)=ASTER(LOC+9)-W(2)
VW(3)=ASTER(LOC+10)-W(3)
WN=(U-W)*(V-W)
W(1)=UW(2)*VW(3)-UW(3)*VW(2)
W(2)=UW(3)*VW(1)-UW(1)*VW(3)
W(3)=UW(1)*VW(2)-UW(2)*VW(1)
D=WV(1)*WN(1)+WV(2)*WN(2)+WV(3)*WN(3)

Determine IF RAY PARALLEL TO PLANE OF TRIANGLE

IF(ABS(D).LE.0.0001)GOTO 490

FIG. 95. (Contd.)
C

DALPHA = (W-XR) * (WB X (V-W))
DALPHA = WXR(1) * (WB(2) + VXR(3) - WB(3) * VXR(2))
1  + WXR(2) * (WB(3) * VXR(1) - WB(1) * VXR(3))
2  + WXR(3) * (WB(1) * VXR(2) - WB(2) * VXR(1))

ALPHA = DALPHA / D
IF (ALPHA < 0) AND (ALPHA < 0) GOTO 40

DBETA = (W-XB) * (U-W) X WB
DBETA = WXB(1) + (UW(2) + WB(3) - UW(3) * WB(2))
1  + WXB(2) + (UW(3) * WB(1) - UW(1) * WB(3))
2  + WXB(3) + (UW(1) * WB(2) - UW(2) * WB(1))

BETA = DBETA / D
IF (BETA < 0) AND (BETA < 0) GOTO 40

GAMMA = 1 - ALPHA - BETA
IF (GAMMA < 0) AND (GAMMA < 0) GOTO 40

COMPUTE DISTANCE TO INTERSECT WITH TRIANGLE

DS = (W-XR) * WN

DS = WXR(1) * WN(1) + WXR(2) * WN(2) + WB(3) * WN(3)
S = DS / D

CALL UNIT (WN)

DIRECT NORMAL INTO ARS FOR ENTRY INTERSECT, OUT OF ARS FOR EXIT INTERSECT

IF (IT - (IT/2) * 2, EQ, 0) GOTO 410
W(1) = WN(1)
W(2) = WN(2)
W(3) = WN(3)
D = 0

410 JSURF = IT
IF (0 LT, 0, 0) JSURF = JSURF

COMPARE NEW INTERSECT DISTANCE WITH DISTANCES ALREADY IN HIT TABLE

STORE HITS (LARGEST TO SMALLEST)

IF (NHIT EQ, 0) GOTO 430
DO 420 I = 1, NHIT
IF (ABS(S-HIT(I)) < 0.0001) GOTO 470
IF (S GT, HIT(I)) GOTO 450

420 CONTINUE

430 NHIT = NHIT + 1
IF (NHIT LE, 20) GOTO 440
WRITE (A, 901) NRO
WRITE (6, 910) (HIT(I), ISURF(I), I = 1, NHIT)
NHIT = 0
GOTO 700

FIG. 95. (Contd.)
IF NEW INTERSECT, STORE HIT

440 HIT(I)=S
  ORML(1+1)=WN(1)
  ORML(2+1)=WN(2)
  ORML(3+1)=WN(3)
  ISURF(I)=JSURF
  GOTO 490

ADD A HIT TO TABLE WHEN S.GT.HIT(I)

450 J=NHIT
  NHIT=NHIT+1
  IF(NHIT.GT.20)GOTO 435

460 IF(J.LT.I)GOTO 440
  HIT(J+1)=HIT(J)
  ORML(1+J+1)=ORM(1+J)
  ORML(2+J+1)=ORM(2+J)
  ORML(3+J+1)=ORM(3+J)
  ISURF(J+1)=ISURF(J)
  J=J-1
  GOTO 460

TWO ENTRIES IDENTICAL WHEN S.EQ. HIT(I)
  IF BOTH RIN OR BOTH HOUT IGNORE
  IF ONE A RIN AND OTHER A HOUT DELETE ENTRY IN TABLE

470 IF(JSURF.ISURF(I).GT.0)GOTO 490

DELETE ENTRY

480 IF(I.GT.NHIT)GOTO 490
  HIT(I)=HIT(I+1)
  ORML(1+I)=ORM(1+I+1)
  ORML(2+I)=ORM(2+I+1)
  ORML(3+I)=ORM(3+I+1)
  ISURF(I)=ISURF(I+1)
  I=I+1
  GOTO 480

INCREMENT TO TEST NEXT POSSIBLE TRIANGLE

490 LOC=LOC+4
499 CONTINUE

ALL POSSIBLE TRIANGLES EXAMINED
  CHECK FOR AN EVEN NUMBER OF HITS

IF(NHIT.NE.0)GOTO 700
IF(NHIT-(NHIT/2)*2.EQ.0)GOTO 500

FIG. 95. (Contd.)
ERROR - INCORRECT SEQUENCE OF HITS

WRITE(6,902)NH0,NHIT
WRITE(6,910)HIT(I),ISURF(I),I=1,NHIT
NHIT=0
GOTO 700

CHECK FOR CORRECT SEQUENCE OF EXITS(•) AND ENTRANCES(•)

500 DO 520 I=2,NHIT
     IF(ISURF(I-1)>ISURF(I),GT,0)GOTO 525
520 CONTINUE
     GOTO 530

ERROR - INCORRECT SEQUENCE OF HITS

525 WRITE(6,903)NH0,NHIT
WRITE(6,910)HIT(I),ISURF(I),I=1,NHIT
NHIT=0
GOTO 700

LOCATE NEXT ROUT DISTANCE RELATIVE TO CURRENT POSITION OF XP

530 IF(HIT(NHIT-1)>0,GOT0 540
     NHIT=NHIT-2
     IF(NHIT.LE.0)GOTO 700
     GOTO 530

CHECK DIRECTION OF NORMAL FOR LARGEST DISTANCE IN HIT TABLE
VERIFY THAT NORMAL IS AN EXIT FOR THE ROUT INTERSECT

540 IF(NASC.EQ.-2)GOTO 600
     IF(ISURF(I),LT,0)GOTO 560
     DO 550 I=1,NHIT
     NORMAL(1,I)=-NORMAL(1,I)
     NORMAL(2,I)=-NORMAL(2,I)
     NORMAL(3,I)=NORMAL(3,I)
     ISURF(I)=ISURF(I)
     550 CONTINUE

STORE HIT TABLE IN ASTER ARRAY
UNLESS COMPUTING NORMAL DISTANCE

560 LOC=LOCHTS
     DO 570 I=1,NHIT
     ASTER(LOC)=HIT(I)
     ASTER(LOC+1)=NORMAL(1,I)
     ASTER(LOC+2)=NORMAL(2,I)
     ASTER(LOC+3)=NORMAL(3,I)
     LOC=LOC+4
     570 CONTINUE

FIG. 95. (Contd.)
CHOOSE CORRECT RIN AND ROUT SET FOR CURRENT POSITION OF XP
   THIS SECTION IS ALSO USED BY REENTRY ROUTINE

600 IF(NHIT.EQ.0)GOTO 700
   RIN=NHIT(NHIT)
   ROUT=NHIT(NHIT-1)
   LAT=1
   LRO=1
   NHIT=NHIT-2
   IF(ABS(DIST-ROUT)*LE.0.0001)GOTO 600
   IF(DIST.GE.ROUT)GOTO 600
   IF(ABS(RIN-ROUT)*LE.0.0001)GOTO 600
   IF(RIN.GT.0.0001)GOTO 800
   RIN=PINF
   LAT=0
   GOTO RAN

700 RIN=PINF
   ROUT=PINF
   LAT=1
   LRO=0
   IF(N.EQ.0)NHIT=0

800 IF(NASC.NE.-2)MASTER(LOCARS+1)=NHIT
   RETURN
   END

FIG. 95. (Concluded)
SUBROUTINE TOR
DIMENSION COEF(4)*RT(4)*XN(3)*XBC(3)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/PAREM/XB(3)*WB(3)*IR
COMMON/GEOM/LBASE,RTN,ROUTE,LR0,PINF,IERR,DIST
COMMON/UNCSTM/NTRIP,NSCAL,NBODY,NRMAX,LTRIP,LOCAL,LRED0,
1 LO'=>['(L)'] ROLE,LOCDA=115*130*LBODY,NA-ST*KNUMBER
COMMON/DAVIS/GRID,LOOPLIST/NORM
EQUIVALENCE (ASTER,MASTER)

CHECK FOR PREVIOUS ENTRY
IF (LOOP.NE.KLOOP) GOTO 10

THIS IS A REENTRY
NR=MASTER(LOCDA+2)
IF (NR.LE.2) RETURN
RIN=ASTER(LOCDA+3)
ROUTE=ASTER(LOCDA+4)
NR=0
GOTO 400

RETRIEVE LOCATIONS OF TORUS DATA
10 CALL UN2(LOCDA,IV,IN)
           LOC=LOCDA+1
           CALL UN2(LOC,IR1,IR2)

            COMPUTE INTERMEDIATE VARIABLES NEEDED TO
            FIND COEFFICIENTS OF QUATRIC EQUATION

            XRC(1)=AH(1)-ASTER(IV)
            XRC(2)=XB(2)-ASTER(IV+1)
            XRC(3)=XB(3)-ASTER(IV+2)
            XN(1)=ASTER(IN)
            XN(2)=ASTER(IN+1)
            XN(3)=ASTER(IN+2)
            R1=ASTER(IR1)
            R2=ASTER(IR2)
            IF (NASC*.NE..-2) GOTO 20
            RS=ABS(R1)
            GOTO 30
20     RSAVE=ABS(DOT(XBC,WH))=R1=R2-R2
            XBC(1)=XBC(1)*RSAVE*WR(1)
            XBC(2)=XBC(2)*RSAVE*WR(2)
            XBC(3)=XBC(3)*RSAVE*WR(3)
30     WDN=DOT(WB,XN)
            XBCD=DOT(XBC,WR)
            XBCD=DOT(XBC,XN)
            XBC*XBCDOT(XBC*XHC)
            R1SO=R1*R1
            R2SO=R2*R2
            TERM=XBC*XHC=R1SO-R2SO

            COMPUTE COEFFICIENTS

            CNEF(1)=4.*XBCD
            CNEF(2)=4.*R1SQ*WDN*WDN+4.*XBCD*XBCD+2.*TERM

FIG. 96. Source Listing, Subroutine TOR
C6 DETERMINE IF 0, 2, OR 4 ROOTS
C7 TWO ROOTS
C8
100 IF(ABS(RT(1)-RT(2)) GT.0.0001) GOTO 110 NR=0
GOTO 500
110 RT(1)=RT(1)+RSAVE
RT(2)=RT(2)+RSAVE
IF(RT(1) LT RT(2)) GOTO 300
T=RT(1)
RT(1)=RT(2)
RT(2)=T
GOTO 300
C9 FOUR ROOTS
C0 DO 210 I=1,4
RT(I)=RT(I)+RSAVE
210 CONTINUE
CO SORT ROOTS IN ASCENDING ORDER
220 IF(RT(1) LE RT(2)) GOTO 230
T=RT(1)
RT(1)=RT(2)
RT(2)=T
230 IF(RT(2) LE RT(3)) GOTO 240
T=RT(2)
RT(2)=RT(3)
RT(3)=T
GOTO 220
240 IF(RT(3) LE RT(4)) GOTO 250
T=RT(3)
RT(3)=RT(4)
RT(4)=T
GOTO 230
C10 IF RAY TANGENT TO SURFACE ELIMINATE INTERSECTS
250 IF(ABS(RT(2)-RT(3)) GT.0.0001) GOTO 260 NR=NR=2
RT(2)=RT(4)
GOTO 270
260 IF(ABS(RT(3)-RT(4)) GT.0.0001) GOTO 270 NR=NR=2
270 IF(ABS(RT(1)-RT(2)) GT.0.0001) GOTO 280 NR=NR=2
RT(1)=RT(3)
RT(3)=RT(4)
280 IF(NR LE 0) GOTO 500
IF(RT(2) GT 0) GOTO 300
NR=NR=2

FIG. 96. (Contd.)
RT(1)=RT(3)  
RT(2)=RT(4)  
GOTO 280

300 IF(NR=2)S00,350,310

C01 FOUR INTERSECTS, DETERMINE WHICHER RIN/ROUT SET REQUIRED

310 IF(ABS(DIST-RT(2))=LE.0.0001)GOTO 320
   IF(DISTLT,RT(2))GOTO 330

320 RIN=RT(3)
   ROUT=RT(4)
   NR=0
   GOTO 400

330 ASTER(LOCDA+3)=RT(3)
   ASTER(LOCDA+4)=RT(4)

350 RIN=RT(1)
   ROUT=RT(2)

400 LRI=1
   LAO=1
   IF(RINGE,ROUT)GOTO 500
   IF(ABS(RIN-ROUT)=LE.0.0001)GOTO 500
   IF(ROUTLE.0.0)GOTO 500
   IF(RINGT.0.0001)GOTO 600
   RIN=+0.0001

500 LRI=0
   GOTO 600

C02 RAY MISSES FROM PRESENT ORIGIN

500 RIN=PINF
   ROUT=PINF
   LRI=0
   LAO=0

600 MASTER(LOCDA+2)=NR
   RETURN
END

FIG. 96. (Concluded)
SUBROUTINE QRTIC(C, R, N)
DIMENSION C(4), R(4), N(3), RR(3)
*
SOLVES A POLYNOMIAL EQUATION OF THE TYPE \( x^4 + C(1) x^3 + C(2) x^2 + C(3) x + C(4) = 0 \) USING THE FERRARI SOLUTION OF
THE QUARTIC EQUATION. THE COEFFICIENT OF \( x^3 \) IS ASSUMED TO BE 1.
R(1) CONTAINS THE ROOTS. N CONTAINS THE NUMBER OF REAL ROOTS.
IF THERE ARE 2 REAL ROOTS THEY WILL BE IN R(1) AND R(2). WITH THE
COMPLEX ROOTS IN R(3) = R(4)*I. IF THERE ARE NO REAL ROOTS
THE COMPLEX ROOTS ARE IN R(1) = R(2)*I AND R(3) = R(4)*I.
*
COMPUTE RESOLVENT CUBIC
*
C15Q = C(1) * C(1)
CC(1) = - 5 * CC(2)
CC(2) = 25 * CC(1) * CC(3) * CC(4)
CC(3) = - 125 * (C(4) * (4 * C(2) - C15Q) * C(1) * CC(3))
CALL CUBIC(CC+RR+NN)
*
DETERMINE IF POSSIBLE SOLUTION
*
TN = 25 * C15Q = C(2)
DO 10 I = 1, NN
ROOT = R(1)
ASQ = TROOT + TROOT
IF (ABS(ASQ) * LE.0.000001) ASQ = 0.
IF (ASQ .LT. 0.0) GOTO 10
RSQ = TROOT + TROOT
IF (ABS(RSQ) * LE.0.000001) RSQ = 0.
IF (RSQ .GE. 0.0) GOTO 20
10 CONTINUE
N = 0
RETURN
*
COMPUTE FIRST TWO ROOTS OF QUARTIC EQUATION
*
20 TWOAB = C(1) * ROOT + C(3)
A = SQRT(ASQ)
B = SIGN(SQRT(RSQ), TWOAB)
N = 0
REAL = 25 * (A + A - C(1))
DISC = REAL * REAL - ROOT * B
SRROOT = SQRT(ABS(SDISC))
IF (ABS(SDISC) * LE.0.000001) DISC = 0.
IF (DISC .LT. 0.0) GOTO 30
*
DISCRIMINATE .GE. 0 COMPUTE 2 REAL ROOTS
*
N = 2
R(1) = REAL + SRROOT
R(2) = REAL - SRROOT
GOTO 40
*
DISCRIMINATE .LT. 0 COMPUTE 2 IMAGINARY ROOTS
*
30 R(3) = REAL
R(4) = SRROOT
*
COMPUTE LAST TWO ROOTS OF QUARTIC EQUATION
*
FIG. 97. Source Listing, Subroutine QRTIC

696
40 REAL=REAL-A
DISC=REAL*REAL-ROOT=0
SQREROOT=SQRT(ABS(DISC))
IF (ABS(DISC) .LE. 0.000001) DISC=0.
IF (DISC.LT.0.0) GOTO 50

C C
C DISCRIMINATE .GE. 0 COMPUTE 2 REAL ROOTS
C
N=N+2
R(N)=REAL-SQREROOT
R(N-1)=REAL+SQEROOT
RETURN

C C
C DISCRIMINATE .LT. 0 COMPUTE 2 IMAGINARY ROOTS
C
50 R(N+1)=REAL
R(N+2)=SQREROOT
RETURN
END

FIG. 97. (Concluded)
SUBROUTINE CUBIC(C,R,N)
DIMENSION C(3),R(3)

COMPUTE ROOTS OF CUBIC EQUATION

C1
C1SQ=C(1)*C(1)
P=C(2)-C1SQ/3.
Q=C(3)+C(1)*(2*C1SQ/27.-C(2)/3.)
DISC4=P*P+P*27.+Q*Q
C3=C(1)/3.
IF(ABS(DISC).LE.1.E-6)DISC=0.
IF(DISC.LE.0.0)GOTO 10

CONDITION FOR 1 REAL AND 2 COMPLEX ROOTS

N=1
SQROOT=SQRT(DISC/108.)
HALFQ=.5*Q
ACU=-HALFQ*SQROOT
BCU=HALFQ*SQROOT
A=SIGN(ABS(ACU))**.3333333333333+ACU)
B=SIGN(ABS(BCU))**.3333333333333+BCU)
AB=A*B
R(1)=AB-C3
R(2)=-.5*AB-C3
R(3)=.666025404*(A-B)
RETURN

CONDITION FOR 3 REAL ROOTS

N=3
T=SQRT(ABS(P)/3.)
TT=T*T
IF(DISC.EQ.0.0)GOTO 20
PHI3=ATAN2(SQRT(-DISC/27.),Q)/3.
R(1)=TT*COS(PHI3)+C3
R(2)=TT*COS(PHI3+2.094395103)+C3
R(3)=TT*COS(PHI3+2.094395103)+C3
RETURN

CONDITION FOR 2 OR 3 EQUAL ROOTS

R(1)=SIGN(T,T)=C3
R(2)=SIGN(T,T)=C3
R(3)=R(2)
RETURN
END

FIG. 98. Source Listing, Subroutine CUBIC
SUBROUTINE UN2(L,J1,J2)
UNPACK 2 15-BIT INTEGER DATA ITEMS FROM L WORD IN MASTER ARRAY

COMMON MASTER(10000)
I3=MASTER(L)
J1=I3/32768
J2=I3-J1*32768
RETURN
END

FIG. 99. Source Listing, Subroutine UN2
SUBROUTINE UN3(L,J1,J2,J3)

UNPACK 2 6-BIT AND 1 15-BIT INTEGER DATA ITEMS FROM G1 WORKING STORAGE AT THE L WORD IN THE MASTER ARRAY

COMMON MASTER(10000)
I3=MASTER(L)
I2=I3/32768
J1=I2/64.
J2=I2*64
J3=I3*32768
RETURN
END

FIG. 100. Source Listing, Subroutine UN3
SUBROUTINE OPENK(L,J1,J2,J3)
COMMON/STRACK/D1,D2,KHIT,LMAX,TR(200),XS(3),IRSTR,JENC.
I   ITR(200),CA,CE,SA,SE
UNPACK 3 12-BIT INTEGER DATA ITEMS FROM COMPONENT LINE-OF-SIGHT
STORAGE ARRAY ITR, THE THREE ITEMS ARE / SURFACE NUMBER / BODY NUMBER / NEXT REGION /
I3=ITR(L)
I2=I3/4096
J1=I2/4096
J2=I2-J1*4096
J3=I3-I2*4096
RETURN
END

FIG. 101. Source Listing, Subroutine OPENK
FUNCTION RAN(M)  
COMMON/RANDM/IRN  
GIVEN A NUMBER IRN, GENERATE A RANDOM NUMBER BETWEEN 0 AND 1  
RAN=URAN31(IRN)  
RETURN  
END  

FIG. 102. Source Listing, Subroutine RAN
FUNCTION URAN31( )
A = 6710886.0
RETURN
END
SUBROUTINE CROSS (ANSWER, FIRST, SECOND)
DIMENSION ANSWER(3), FIRST(3), SECOND(3)
COMPUTE CROSS PRODUCT ANSWER = FIRST X SECOND
ANSWER(1) = FIRST(2) * SECOND(3) - FIRST(3) * SECOND(2)
ANSWER(2) = FIRST(3) * SECOND(1) - FIRST(1) * SECOND(3)
ANSWER(3) = FIRST(1) * SECOND(2) - FIRST(2) * SECOND(1)
RETURN
END

FIG. 104. Source Listing, Subroutine CROSS
FUNCTION DOT(FIRST, SECOND)
DIMENSION FIRST(3), SECOND(3)

COMPUTE DOT PRODUCT
DOT = FIRST + SECOND
DOT = FIRST(1) * SECOND(1) + FIRST(2) * SECOND(2) + FIRST(3) * SECOND(3)
RETURN
END

FIG. 105. Source Listing, Subroutine DOT
SUBROUTINE UNIT(V)
DIMENSION V(3)

* Compute unit vector (direction cosines of vector)
  
  TEMP = SQRT(DOT(V,V))
  V(1) = V(1)/TEMP
  V(2) = V(2)/TEMP
  V(3) = V(3)/TEMP
RETURN
END

FIG. 106. Source Listing, Subroutine UNIT
FUNCTION XDIST(XA,XB)

COMPUTE THE DISTANCE BETWEEN TWO GIVEN POINTS XA AND XB

DIMENSION XA(3),XB(3)
XSUM=0,
DO 10 I=1,3
XSUM=XSUM+(XA(I)-XB(I))**2
10 CONTINUE
XDIST=SQRT(XSUM)
RETURN
END

FIG. 107. Source Listing, Subroutine XDIST
SUBROUTINE DCOSP(XA,XB,WA)

COMPUTE THE DIRECTION COSINES FROM POINT XA TO POINT XB
AND STORE DIRECTION COSINES IN WA

DIMENSION XA(3),XB(3),WA(3)

DIS= XDIST(XA,XB)

DO 10 I=1,3
  WA(I)=(XB(I)-XA(I))/DIS

10 CONTINUE

RETURN

END

FIG. 108. Source Listing, Subroutine DCOSP
SUBROUTINE TROPIC(WP)

GENERATE RANDOM DIRECTION COSINES FROM AN ISOTROPIC DISTRIBUTION

DIMENSION WP(3)
10 X1=ran (-1)
   X2=ran,(-1)
   X1S=X1^2
   X2S=X2^2
   T=X1S*X2S
   IF(T.GE.1.)GOTO 10

COMPUTE THE SINE AND COSINE OF A RANDOM ANGLE PHI

CSPHI=(X1S-X2S)/T
SNPHI=(2.*X1*X2)/T
X1=ran (-1)
IF(X1.LE.-5.)SNPHI=SNPHI

COMPUTE THE SINE AND COSINE OF A RANDOM ANGLE THETA

CSTHT=2.*ran (-1)-1,
SNHT=sqrt(1.-CSTHT**2)

COMPUTE RANDOM DIRECTION COSINES

WP(1)=SNHT*CSPHI
WP(2)=SNHT*SNPHI
WP(3)=CSTHT
RETURN
END

FIG. 109. Source Listing, Subroutine TROPIC
FUNCTION S(I,N)
DIMENSION MASTER(10000)
COMMON ASTER(10000)
COMMON/GEOM/LBASE, RIN, ROUT, LRI, LRO, PINF, IERR, DIST
EQUVALENCE (MASTER,ASTER)

S RETRIEVES COORDINATE OF ANY ONE OF THE 6 SIDES OF AN RPP
I IS THE RPP NUMBER     N IS THE SURFACE NUMBER

L=LBASE+12*(I=1)+2*(N=1)
LL=MASTER(L+1)
S=ASTER(LL)
RETURN
END

FIG. 110. Source Listing, Subroutine S
SUBROUTINE RPP2(LSURF, XP, IRP)

FIND NUMBER OF ABUTTING RPP TO INTERSECTED SURFACE

DIMENSION XP(3)
COMMON ASTER(10000)
COMMON/PAREM/XB(3),WB(3),IR
COMMON/BEO/NMBASE,LRIN,LR0,LR0,PINF,IRRR,DIST
COMMON/UNCGEM/NRP,PNTRIP,NSCAL,NBODY,NMAX,NTRIP,LSURF,LREAD
1 LOCAT=LRIN,LR0,LOCDA,N3,LOCAL30=130,NBODY,NSCAL,LOOP

LOC=LOCA*120(NASC=1)=2(LSURF+1)
CALL UN2(LOC,LOCAT,NC)
IF(NC=1)10,20,30
10 IRP=0
RETURN
20 CALL UNZ(LOC,LOCAT,NC)
RETURN
30 M=1

DO 90 I=1,NC
M=M+1
IF(M.GT.0)GOTO 50
CALL UN2(LOCAT,I1,I2)
LOCAT=LOCA*1
IRP=I1
GOTO 70
50 IRP=I2
70 LS=(1-LSURF)/2
DO 80 J=1,3
IF(J.EQ.LS)GOTO 80
IF((S(IRP,2*J)=XP(J),(XP(J)=S(IRP,2*J))<LT.0)GOTO 90
80 CONTINUE
RETURN
90 CONTINUE
IRP=0
RETURN
END

FIG. 111. Source Listing, Subroutine RPP2
SUBROUTINE VOLUM

COMPUTE VOLUMES BY REGION IN VOLUME DEFINED BY BOX

* DIMENSION VASTER(1000), WTB(3), WOB(3), DSP(3),
  1 XV(3), XT(3), XA(3), XO(3), XP(3), XTEMP(3)

* COMMON ASTER(10000)
  COMMON/PERIM/XH(3), WBT(3), IR
  COMMON/GEOM/LBASE, RIN, ROUT, LRI, LR0, PIN, FERR, DIST
  COMMON/UNCERT/NRP, NTRIP, NSCAL, NBODY, NMRAX, LTRIP, LSCAL, LREGO,
  1 LDATA, LRAIN, LROT, LIG, LOCDA, IIG, I30, LBODY, NAPC, KLOOP

* COMMON/WCT, LIRFO, NG1ERR

901 FORMAT (3E20.8)
902 FORMAT (2E20.8)
903 FORMAT (1H0, 1X, 6HVERTEX, 14X, 6HTOP, 14X, 6HSIDE, 1PT)
904 FORMAT (4E20.8)
905 FORMAT (1H0, 8X, 2E20.8)
906 FORMAT (2E20.8)
907 FORMAT (1H0, 1X, 6HVASTER OVERWRITE, 5X, 6HNMAX = I5)
908 FORMAT (1H0, 1X, 6HSTARTING REGION IS, IS)
909 FORMAT (1H0, 1X, 6HWRITE OVERWRITE, 5X, 6HNMAX = I5)
910 FORMAT (1H0, 1X, 6HREAD CARD/110/INDEX NOT PROCESSED)
911 FORMAT (1H0, 1X, 6HREAD CARD/110/INDEX NOT PROCESSED)
912 FORMAT (1H0, 1X, 6HREAD CARD/110/INDEX NOT PROCESSED)
913 FORMAT (1H0, 1X, 6HREAD CARD/110/INDEX NOT PROCESSED)

READ (5, 906) IR, N0; ERR
IF (NOERR .LE. 0) N0; ERR = 25

ENTER COORDINATES OF BOX

READ (5, 901) (XV(I), I=1, 3)
READ (5, 901) (XT(I), I=1, 3)
READ (5, 901) (XO(I), I=1, 3)
READ (5, 901) (XA(I), I=1, 3)

ENTER CELL SIZE

READ (5, 902) DOD, DT
WRITE (6, 903)
WRITE (6, 904) (XV(J), XT(J), XO(J), XA(J), J=1, 3)
WRITE (6, 905) DOD, DT
WRITE (6, 908) IR
IF (NMRAX .GT. 2000) WRITE (6, 909) NMRAX
CALL DCOSP (XV, XT, WTB)
CALL DCOSP (XV, XO, WOB)
CALL DCOSP (XV, XA, WAB)
XOIS = XOIST (XV, XA)
TEST0 = 0
TEST1 = 0
XTEMP(1) = 0
DO 10 I=1, NMRAX
  VASTER(I) = 0
  JR = IR
  IR = JR
10 CONTINUE

COMPUTE NUMBER OF HORIZONTAL AND VERTICAL CELLS

FIG. 112. Source Listing, Subroutine VOLUM
N2=ROUND((XX+XO)/DOD+1.0)
N1=ROUND((XX+XT)/DT+1.0)

TRACE RAYS FROM LOWER RIGHT CORNER OF EACH CELL

DO 300 J=1,N2
DO 100 I=1,N1
DSP(I)=WT(I)*DT
XP(I)=XX(I)
WB(I)=WA(I)
100 CONTINUE
S1=0
IRA=JIR

TRACE ALL RAYS FROM COLUMN OF CELLS

DO 200 I=1,N1
NHA=1

TRACE RAY THROUGH BOX VIA SUBROUTINE G1

110 CALL G1(S1,IRPRIM,XP)
IF(IERR.0E.-NG1ERR)GOTO 400
VASTER(IR)=VASTER(IR)+S1
IF(DIST.0E.-XVDIS)GOTO 115
IF(IRPRIM.LE.0)GOTO 120
IRA=IRPRIM
GOTO 110
115 VASTER(IR)=VASTER(IR)-(DIST-XVDIS)
120 XTEMP(1)=WB(1)
XTEMP(2)=WB(2)
XTEMP(3)=WB(3)
IRA=JIR
TESTDN=TESTDN+DT
IF(TESTDN.GT.0)GOTO 180
WB(1)=WTB(1)
WB(2)=WTB(2)
WB(3)=WTB(3)
NASC=1

DETERMINE REGION OF NEXT ORIGIN OF RAY IN COLUMN

CALL G1(S1,IRPRIM,XP)
IF(IERR.0E.-NG1ERR)GOTO 400
IF(S1-0.0)130*160*170
130 IRA=IRPRIM
JIR=IR
CALL G1(S1,IRPRIM,XP)
IF(IERR.0E.-NG1ERR)GOTO 400
IF(DIST-0.0)140*160*170
140 IRA=IRPRIM
STOP
160 IRA=IRPRIM
JIR=IR
170 TESTDN=S1

SHIFT ORIGIN OF RAY TO NEXT CELL IN COLUMN

180 DO 190 JI=1,3
WB(JI)=XTEMP(JI)
190 CONTINUE
FIG. 112. (Contd.)

```
XBJ (I) = XB (J) + DSP (JI)
190 CONTINUE
200 CONTINUE
C
C ONE COLUMN OF CELLS COMPLETE = SHIFT TO NEXT COLUMN

210 NASEC = I
DO 220 I = 1, 3
WB (I) = WOB (I)
XB (I) = XV (I)
220 CONTINUE
JIR = IRJ
IA = JIR
TESTOD = 0
TESTOV = TESTOV = DD
IF (TESTOV) 230 + 230 + 280
C
C DETERMINE REGION OF FIRST ORIGIN OF NEXT COLUMN

230 CALL G1 (S1, IRPRIM, XP)
IF (ERR .GE. NSF1ERR) GOTO 400
IF (S1 = DD) 240 + 260 + 270
240 IRPRIM = IRJ
IF (IRPRIM) 250 + 280 + 270
250 IF (DIST = DD) 250 + 260 + 270
255 STOP
260 IRPRIM = IRJ
270 TESTOV = S1
C
C SHIFT ORIGIN OF NEXT RAY TO FIRST ORIGIN OF NEXT COLUMN OF CELLS

280 DO 290 I = 1, 3
XAI (I) = XA (I) + XOB (I) + DD
XV (I) = XV (I) + WOB (I) + DD
XT (I) = XT (I) + WOB (I) + DD
290 CONTINUE
JIR = IRJ
300 CONTINUE
C
C ALL RAY DISTANCES THROUGH EACH REGION IN BOX ACCUMULATED

310 READ (5, 910) IR1, VR
IF (ERR .GE. NSF1ERR) GOTO 500
IF (IR1 .LE. 0) IR1 = NRMAX * I
SUMVR = 0.
C
C COMPUTE VOLUME OF EACH REGION IN BOX

400 DO 450 I = 1, NRMAX
VASTER (I) = VASTER (I) + DD * DT
IF (I = IR1) 410 * 430 * 420
410 WRITE (6, 9101) VASTER (I)
GOTO 440
420 WRITE (6, 911) IR1, VR
READ (5, 910) IR1, VR
GOTO 410
```

714
**FIG. 112. (Concluded)**

```plaintext
C C C
C 6   COMPUTE PERCENT ERROR FOR PRE-COMPUTED VOLUME OF GIVEN REGION
C
C 430 XPERC=100.* (VASTER(I)/VR-1.),
C     WRITE (6,912)I,VASTER(I),VR,XPERC
C     VASTER(I)=VR
C     READ (5,910)IR1,VR
C C C
C 7   COMPUTE TOTAL VOLUME OF ALL REGIONS WITHIN BOX
C
C 440 SUMV=SUMV+VASTER(I)
C 450 CONTINUE
C     WRITE (6,913)SUMV
C 500 IERR=0
C     RETURN
C     END
C
C
```

715
SUBROUTINE AREA
DIMENSION XP(3), WP(3), XBS(3), CONVRT(4*6), TYPEUN(6)
COMMON ASTER/(10000)
COMMON/PAREM/XB(3), WB(3), IR
COMMON/GEOM/LBAE$=MIN, ROUT=LRISTRO, LRI=LRSTRO, PINF, IERR, DIST
COMMON/UNCSEM/NAPP, NTRIP, NSCAL, NBODY, NRMAX, LRIP, LSCAL, LREGO,
1 LDATA=LRI, LRO=T, ILOCDA=15, I36=1, BODY=NASCR, KLOOP
COMMON/CAL/NIR, SLOS, ANGLE, NTYPE, SSSPACE, LXS(3), MS(3),
1 TRAVEL, SNV, V, H, IVIM
COMMON/WALT/LIRFO, N51ERR
COMMON/CLEAR/CESIZE
COMMON/ENG/LEOMET

C 901 FORMAT(10+6X+2A2)
902 FORMAT(6E17.5)
906 FORMAT(1H0=22M MEMORY OVERLAP IN AREA=5X,7H=LEOMET=,10X,
1 S,6X=LAREA=,12X=LIRFO=)
909 FORMAT(1H0=13ERROR IN AREA=5X,6H=HICODE = 0)
910 FORMAT(1H0=8H=AZIMUTH=F10.3, 5X,10HELEVATION=F10.3)
911 FORMAT(1H0=12CELL SIZE 15,6F1+1X,1H=6F1+1X,6H=610X).
1 12MAREA IN SO+1X+21X+1H+)
912 FORMAT(1H0=96H=HICODE=19X,4H=HAREA/)
913 FORMAT(156F12.5)
914 FORMAT(1H0=13PRESENTED AREA=F12.5)
915 FORMAT(1H0=14HNUMBER OF CELLS 15+15+10X, 1 22KNUMBER OF CELLS AT 15+15)

C DATA HMIN=HHT, HMC, HMB=HNB, HMBB2MIN=HHT, HMC=HMB+2M, 2H+2H /
TYPEUN(1)=HMIN
TYPEUN(2)=HHT
TYPEUN(3)=HMC
 TYPEUN(4)=HMB
CONVRT(1)=1,
CONVRT(1+1)=0.00694444444444
CONVRT(1+3)=6.451625806
CONVRT(1+4)=0.006451625806
CONVRT(2)=1449,
CONVRT(2+1)=1449,
CONVRT(3)=929, 0341161
CONVRT(2+2)=0.09290341161
CONVRT(3+1)=15499969
CONVRT(3+2)=0.01075386736
CONVRT(3+3)=1,
CONVRT(3+4)=0.0001
CONVRT(3+4)=15499969
CONVRT(4+1)=10+7636736
CONVRT(4+3)=10000,
CONVRT(4+4)=1,
BLANK=HMBB

C COMPUTE AND INITIALIZE AREA FOR STORING PRESENTED AREA
BY COMPONENT CODE

LAREA=LIRFO=1000
IF(LAREA=GE.LEOMET)GOTO 10
WRITE (6+908)LEOMET, LAREA=LIRFO
STOP
10 LAREA=LIRFO=1
DO 20 L=LAREA, LAREA
ASTER(L)=0.

FIG. 113. Source Listing, Subroutine AREA
20 CONTINUE
C READ GRID INPUT PARAMETERS
C READ (5,901) NX,NY,IRSTAT, IENC,NG1ERR,NSTART,NENO,CELLUN,AREAUN
READ (5,902) AE,ENOTH,ISHIFT,GROUND
READ (5,902) XSHIFT,YSHIFT,CELSIZ
C INITIALIZE PARAMETERS NOT SET BY INPUT
C IF (IRSTAT .LE. 0) IIRSTAT=1
IF (CELSIZ .LE. 0.) CELSZ=2
IF (NSTART .LE. 0.) NSTART=1
IF (NG1ERR .LE. 0.) NG1ERR=25
IF (AREAUN.EQ.BLANK) AREAUN=HMIN
IF (CELLUN.EQ.BLANK) CELLUN=HMIN
C DETERMINE MEASUREMENT UNITS AND COMPUTE GRID CELL AREA
C DO 30 I=1,4
IF (CELLUN.EQ.TYPEUN(I)) GOTO 40
CONTINUE
30 DO 50 J=1,4
IF (AREAUN.EQ.TYPEUN(J)) GOTO 60
CONTINUE
50 AREAC=CELSIZ*CELSIZ*CONVAT(I,J)
C RADIAN= .017453292519943
AR=10RADIAN
ER=ERADIAN
SA=SNRADIAN
CA=COS(AR)
SE=SIN(ER)
CE=COS(ER)
KL=NX*NY
NMIT=0
C PROCESS KL CELLS IN GRID PLANE
C DO 200 KK=ISTART,KL
WB(1)=CE*CA
WB(2)=CE*SA
WB(3)=SE
C COMPUTE ROW AND COLUMN NUMBER OF GRID CELL
C II=((KK-1)/NX)+1
J=KK-(II-1)*NX
C CELL2=5*CELSIZ
V=FLOAT((NY/2)-II)*CELSIZ+CELL2
VREF=V*CELL2
H=FLOAT((NX/2)-J)*CELSIZ+CELL2
HREF=H*CELL2
IV=RAN(-1)*10.
IH=RAN(-1)*10.
IVH=10*IH+IV
C COMPUTE RANDOM POINT WITHIN GRID CELL

FIG. 113. (Contd.)
FIG. 113. (Concluded)
SUBROUTINE TESTG

TRACE A RAY BETWEEN TWO GIVEN POINTS XB TO XBF

DIMENSION XP(3), XB(3)
COMMON/PPAR/XB(3), IR
COMMON/GEOM/LBASE, LROUT, LR, LRG, PINF, ERR, DIST
COMMON/UNCGEN/MNP, NTRIP, NSCAL, NBODY, NRM, LTRIP, LSCA, LREG
1 LDATA, LROUT, LRO,T, LIO, LCDA, LST, L0, LBACK, LSCA, LLOOP
COMMON/WALT/LRFO, N01ERR

901 FORMAT(2I10)
902 FORMAT(3I10, 2X, 6I10)
903 FORMAT(19H ENTER NUMBER OF RAYS)
904 FORMAT(19H READ (5, 901)NRAYS, NG1ERR
905 FORMAT(19H WRITE (6, 902)NRAYS
906 FORMAT(19H IF (NG1ERR. LE. 0.) NG1ERR = 25
907 FORMAT(19H ENTER POINT COORDINATES AND REGION OF EACH)
908 FORMAT(19H READ (5, 903)XB, IRSTRT
909 FORMAT(19H READ (5, 903)XBP, IRFIN
910 FORMAT(19H WRITE (6, 904)XB, IRSTRT, XBP, IRFIN
911 FORMAT(19H RANGE = XDIST(XB, XBP)
912 CALL DCOSP(XB, XBP, WB)
913 FORMAT(19H WRITE (6, 905)WB, RANGE
914 FORMAT(19H IRSTRT
915 FORMAT(19H NASC = 1
916 FORMAT(19H WRITE (6, 906)
917 FORMAT(19H TRACE RAY TO NEXT REGION INTERSECT
918 CALL G1(S1, IRPRIM, XP)
919 IF (ERR. GE. NG1ERR) GOTO 60
920 IF (G1(S1, IRPRIM, S1, XP, DIST
921 IF (DIST. GE. RANGE) GOTO 30
922 IF (IRPRIM. LE. 0.) GOTO 20
923 IF (IRPRIM)
924 GOTO 10
925 FORMAT(19H WRITE (6, 908)IR
926 GOTO 50
927 IF (IR. NE. IRFIN) GOTO 20
928 CONTINUE
929 IF (ERR. NE. 0.) RETURN
930 END

FIG. 114. Source Listing, Subroutine TESTG
FIG. 115. Listing, Sample Problem Data Deck
<p>| | | | | | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17</td>
<td>-70</td>
<td>20</td>
<td>15</td>
<td>-10</td>
<td>10</td>
<td>15</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-30</td>
<td>10</td>
<td>15</td>
<td>-30</td>
<td>20</td>
<td>15</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-70</td>
<td>20</td>
<td>15</td>
<td>-70</td>
<td>10</td>
<td>25</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-30</td>
<td>10</td>
<td>25</td>
<td>-30</td>
<td>20</td>
<td>15</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-70</td>
<td>20</td>
<td>15</td>
<td>-70</td>
<td>20</td>
<td>15</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-30</td>
<td>20</td>
<td>35</td>
<td>-30</td>
<td>20</td>
<td>15</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-70</td>
<td>20</td>
<td>35</td>
<td>-30</td>
<td>20</td>
<td>15</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>-30</td>
<td>20</td>
<td>15</td>
<td>-30</td>
<td>20</td>
<td>15</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18 REC</td>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>28</td>
<td>TRUNK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>7.5</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 SPH</td>
<td>0</td>
<td>0</td>
<td>52</td>
<td>5</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20 TEC</td>
<td>-7.5</td>
<td>49</td>
<td>20</td>
<td>0</td>
<td>-12</td>
<td>ARM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>7.5</td>
<td>49</td>
<td>20</td>
<td>0</td>
<td>-12</td>
<td>ARM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>7.5</td>
<td>49</td>
<td>20</td>
<td>0</td>
<td>-12</td>
<td>ARM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22 TRC</td>
<td>-2</td>
<td>-4.5</td>
<td>27</td>
<td>32</td>
<td>0</td>
<td>-12</td>
<td>LEG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23 TRC</td>
<td>-2</td>
<td>-4.5</td>
<td>27</td>
<td>32</td>
<td>0</td>
<td>-12</td>
<td>LEG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24 TCR</td>
<td>21.5</td>
<td>37</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>STEERING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>8</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25 ARB</td>
<td>21.5</td>
<td>6</td>
<td>33.5</td>
<td>21.5</td>
<td>6</td>
<td>33.5</td>
<td>WHEEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>21.5</td>
<td>6</td>
<td>44</td>
<td>40</td>
<td>0</td>
<td>37</td>
<td>STEERING</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>21.5</td>
<td>6</td>
<td>44</td>
<td>40</td>
<td>0</td>
<td>37</td>
<td>WHEEL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>3127</td>
<td>2146</td>
<td>4328</td>
<td>1345</td>
<td>3127</td>
<td>3127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FIG. 115 (Concluded)**