THE DESCRIPTION OF THE ROBIN PROGRAM AND ITS CONVERSION TO THE ETC(U)

MAY 80 M D MERRILL, D ELWELL

DAAD07-76-C-0115

ERADCOM/ASL-CR-80-0115-1
THE DESCRIPTION OF THE ROBIN PROGRAM AND ITS
CONVERSION TO THE INTERDATA 7-32 COMPUTER SYSTEM

LEVEL

By

M. DON MERRILL
DONALD ELWELL

Electrical and Computer Engineering
New Mexico State University
Las Cruces, New Mexico 88003

MAY 1980

Under Contract: DAADO7-76-C-0115

Contract Monitor: BRUCE W. KENNEDY

Approved for public release; distribution unlimited

US Army Electronics Research and Development Command
ATMOSPHERIC SCIENCES LABORATORY
White Sands Missile Range, NM 88002
NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

The citation of trade names and names of manufacturers in this report is not to be construed as official Government endorsement or approval of commercial products or services referenced herein.

Disposition

Destroy this report when it is no longer needed. Do not return it to the originator.
DISCLAIMER NOTICE

THIS DOCUMENT IS BEST QUALITY PRACTICABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.
THE DESCRIPTION OF THE ROBIN PROGRAM AND ITS CONVERSION TO THE INTERDATA 7-32 COMPUTER SYSTEM:

ROBIN (the operational program for computation of upper atmospheric density and other parameters from a passive inflatable falling sphere) was developed for the Air Force Cambridge Research Laboratory by the University of Dayton Research Institute in the 1960's. Its utilization was limited to the Meteorological Rocket Network ranges that had high precision radars and large capacity computers. The development of minicomputers and the deployment of Nike Hercules tracking radars at several Army meteorological rocket sites made it
possible to expand the use of the ROBIN sphere to remote locations. The only remaining barrier was the adaptation of the ROBIN program to a minicomputer. This report describes the conversion of the ROBIN-UNIVAC 1108 program used by the Atmospheric Sciences Laboratory at White Sands Missile Range, New Mexico, to the Interdata 7-32 minicomputer.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>I. DESCRIPTION OF PROGRAM</td>
<td></td>
</tr>
<tr>
<td>A. ROBIN Program</td>
<td>6</td>
</tr>
<tr>
<td>B. Data Editing Program</td>
<td>6</td>
</tr>
<tr>
<td>II. COMPARISON OF INTERDATA AND UNIVAC OUTPUTS</td>
<td>14</td>
</tr>
<tr>
<td>Figure 1. Plot of Winds from UNIVAC and Interdata Systems for Common Benchmark Data</td>
<td>15</td>
</tr>
<tr>
<td>Table 1. Pressure, Temperature and Density Comparison of UNIVAC and Interdata Outputs Utilizing Benchmark Input Data</td>
<td>16</td>
</tr>
<tr>
<td>ROBIN Outputs - Interdata 7-32 Computer</td>
<td>19</td>
</tr>
<tr>
<td>III. CONCLUSION</td>
<td>29</td>
</tr>
<tr>
<td>IV. FLOWCHARTS</td>
<td>31</td>
</tr>
<tr>
<td>General Data Flow for UNIVAC 1108 Computer System</td>
<td>33</td>
</tr>
<tr>
<td>Main Program ROBIN</td>
<td>35</td>
</tr>
<tr>
<td>Subroutine ATOMS</td>
<td>41</td>
</tr>
<tr>
<td>Subroutine BLKDAT</td>
<td>42</td>
</tr>
<tr>
<td>Subroutine CONTROL</td>
<td>43</td>
</tr>
<tr>
<td>Subroutine CORRS</td>
<td>44</td>
</tr>
<tr>
<td>Subroutine DECALTS</td>
<td>45</td>
</tr>
<tr>
<td>Subroutine DEV</td>
<td>46</td>
</tr>
<tr>
<td>Subroutine DRAGT</td>
<td>47</td>
</tr>
<tr>
<td>Subroutine DRIVE</td>
<td>48</td>
</tr>
<tr>
<td>Subroutine DRVTS</td>
<td>49</td>
</tr>
<tr>
<td>Subroutine FITON</td>
<td>50</td>
</tr>
<tr>
<td>Subroutine INTER</td>
<td>51</td>
</tr>
<tr>
<td>Subroutine LEGENDR</td>
<td>52</td>
</tr>
<tr>
<td>Subroutine READPT</td>
<td>53</td>
</tr>
<tr>
<td>Subroutine READ1</td>
<td>54</td>
</tr>
<tr>
<td>Subroutine REAVG</td>
<td>55</td>
</tr>
<tr>
<td>Subroutine REAVGTS</td>
<td>56</td>
</tr>
<tr>
<td>Subroutine RTDATA</td>
<td>57</td>
</tr>
<tr>
<td>Subroutine SINGLE</td>
<td>58</td>
</tr>
<tr>
<td>Subroutine SLIDE</td>
<td>59</td>
</tr>
<tr>
<td>Subroutine SLIDET</td>
<td>60</td>
</tr>
<tr>
<td>Subroutine TAB</td>
<td>61</td>
</tr>
<tr>
<td>Subroutine TABLE</td>
<td>62</td>
</tr>
<tr>
<td>Subroutine THEOT</td>
<td>63</td>
</tr>
<tr>
<td>Subroutine TIFALL</td>
<td>64</td>
</tr>
<tr>
<td>Subroutine TIFAL2</td>
<td>65</td>
</tr>
<tr>
<td>Subroutine TROBIN</td>
<td>66</td>
</tr>
<tr>
<td>Subroutine WANGL</td>
<td>67</td>
</tr>
<tr>
<td>Main Program PMR</td>
<td>68</td>
</tr>
<tr>
<td>Main Program AO</td>
<td>69</td>
</tr>
<tr>
<td>Subroutine MAIN</td>
<td>70</td>
</tr>
<tr>
<td>Subroutine MDECK</td>
<td>71</td>
</tr>
</tbody>
</table>
### TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subroutine READST</td>
<td>82</td>
</tr>
<tr>
<td>Subroutine READTT</td>
<td>83</td>
</tr>
<tr>
<td>Subroutine WRTAPE</td>
<td>85</td>
</tr>
<tr>
<td>V. PROGRAM LISTINGS FOR INTERDATA COMPUTER</td>
<td>86</td>
</tr>
<tr>
<td>Main Program ROBIN</td>
<td>86</td>
</tr>
<tr>
<td>Subroutine ATMOS</td>
<td>97</td>
</tr>
<tr>
<td>Subroutine BLKDTA</td>
<td>101</td>
</tr>
<tr>
<td>Subroutine CONTRL</td>
<td>102</td>
</tr>
<tr>
<td>Subroutine CORRS</td>
<td>103</td>
</tr>
<tr>
<td>Subroutine DECALT</td>
<td>104</td>
</tr>
<tr>
<td>Subroutine DEV</td>
<td>105</td>
</tr>
<tr>
<td>Subroutine DRAGT</td>
<td>107</td>
</tr>
<tr>
<td>Subroutine DRVT</td>
<td>112</td>
</tr>
<tr>
<td>Subroutine FITON</td>
<td>113</td>
</tr>
<tr>
<td>Subroutine INTER</td>
<td>114</td>
</tr>
<tr>
<td>Subroutine LINEAR</td>
<td>116</td>
</tr>
<tr>
<td>Subroutine REAVG</td>
<td>117</td>
</tr>
<tr>
<td>Subroutine REAVGT</td>
<td>118</td>
</tr>
<tr>
<td>Subroutine SLIDE</td>
<td>119</td>
</tr>
<tr>
<td>Subroutine SLIDET</td>
<td>120</td>
</tr>
<tr>
<td>Subroutine TAB</td>
<td>121</td>
</tr>
<tr>
<td>Subroutine THEOT</td>
<td>123</td>
</tr>
<tr>
<td>Subroutine TIFAIL</td>
<td>124</td>
</tr>
<tr>
<td>Subroutine TIFAL2</td>
<td>128</td>
</tr>
<tr>
<td>Subroutine TROBIN</td>
<td>132</td>
</tr>
<tr>
<td>Subroutine WANGL</td>
<td>136</td>
</tr>
<tr>
<td>VI. REFERENCES</td>
<td>137</td>
</tr>
</tbody>
</table>
INTRODUCTION

This report describes the results of converting the 1977 ROBIN* program written for the UNIVAC 1108 computer (36 bit) to a ROBIN program written for the Interdata 7/32 computer (32 bit).

In making the conversion, it was necessary to reduce the physical size of the resulting object program and limit the number of input/output units used by ROBIN because of the limited memory of the Interdata computer (128,000 bytes) and limited input/output units (1 magnetic tape unit).

A flow chart for each ROBIN subroutine was made and is included so that any unnecessary calculations or programs could be eliminated. The results of these flow charts indicate that the subroutine LEGNDR, which is a very large program, (35,000 bytes) can be replaced by data statements for the cubic coefficients PXY1(51), PXY2(35), and PZ2(21). These coefficient values are fixed and do not change during the calculations. Since the linear coefficients change during the flight, a new program LINEAR was written to compute the values of PZ1 for points from 19 to 51.

All data in the subroutines DRAGT and ATMOS have been converted from floating numbers (full word) to half word integers. This procedure reduces by about half the size of these two programs.

Comparative results between the 1108 and Interdata ROBIN programs are included for benchmark data taken from an FPS-16 radar track of a sphere. This data is contained on a unilog tape library locator number Y388 at White Sands Missile Range.

*For a detailed description see references 1 and 2.
I. DESCRIPTION OF PROGRAMS

This section discusses each subroutine and program used by the Interdata system and the UNIVAC 1108 system. Due to the different requirements of the systems some routines are different. The following notation will be observed for all routines:

* Denotes calling argument for the Interdata program
** This program is not used by the Interdata system
*** This program is not used by the 1108 system.

A. ROBIN Program

ATMOS (HI,T,RHOO)

This program is sent an altitude, HI, (meters) and returns Temperature, T, (degree Kelvin) and density, RHOO. These returned data are found by interpolating into data tables corresponding to the 1976 standard atmosphere.

CONTROL (GS,NZMID,KSWZ,KSW,IP,PE,IV)

Subroutine used to control the calculation of the theoretical trajectory and bias correction values.

CORRS (N,A,M,B,C2,IS)

Subroutine that calculates a correlation value based on the filter coefficients. This constant is used by the subroutine DEV to compute the RMS noise error.
DECALT (JCNT,$,$) - (JCNT,IFLAG1)*

Subroutine reads radar data until there are 5 consecutive decreasing altitudes then selects the next data point corresponding to an even second of time as the initial trajectory point.

DEV (VRHO,VWX,VWY,VP,RHOOO)

Subroutine that calculates the RMS error associated with the winds, pressures, and densities.

DRAGT (CD,AMC,RE,IY,IPGE,$) - (CD,AMC,RE,IFLAG)*

This subroutine is sent a Mach number (AMC), a Reynolds number (RE), from these data a three dimensional table look-up is performed to find the corresponding coefficient of drag (CD).

DRIVE***

Main program used to read unilog tape and writes this data to unit 14 via calls to RTDATA.

DRVT (K,TEMP,RE,AMC,CD,DENTT,HI,LLL)

Subroutine calculates Reynolds number and Mach number based on velocity calculations and temperature and density data from subroutine ATMOS. The drag coefficient is then obtained from subroutine DRAGT.

FITON

Subroutine called by ROBIN and TROBIN to calculate smoothed velocity points by fitting a Legendre polynomial to the raw positional points. Acceleration fits are done
in ROBIN and TROBIN themselves. FITON2 is an entry point within FITON and is used when the number of points for X and Y smoothing is less than the number of points for Z smoothing.

INTER (INTERP, Z2, Z3, IY, IPGE, STALT)

Subroutine that interpolates the one second data calculated by ROBIN for altitude increments of either 200 meters (INTERP ≠ 2) or 1000 meters (INTERP = 2).

LEGNDR (NPTS, MPWR, CF, CFSQ, CF1, CF2, NDER)**

Subroutine called by SINGLE to calculate Legendre polynomial filter coefficients for degrees up to 10 and up to 100 points. For both the January 78 and September 79 ROBIN programs, the cubic coefficients PXY1(51), PXY2(35) and PZ2(21) are fixed. Since these values never change they are stored as data statements in the Interdata program. Since the linear coefficients PZ1 do change they are calculated by subroutine LINEAR in the Interdata program. This approach saves 32,000 bytes of storage.

LINEAR (NPTS)***

Subroutine only used by the Interdata program to calculate the linear filter coefficients PZ1. See description under LEGNDR.

READPT(IX, ICNT, T, S1, S2, S3, S, $)**

Subroutine that returns a single data point of T, X, Y and Z at one tenth second intervals. Program reads data from an array generated by subroutine READ1.
READ1 (IX,D,N,JSTAT)**

Subroutine Reads edited X,Y,Z and T data from unit 15 into a 24 point array.

REAVG (ICNT,$,$) - (ICNT,IFLAG3)*

This subroutine obtains 5 one tenth second data points from READPT and returns the average to the calling program. This program in the Interdata system replaces subroutine READPT and READ1.

REAVGT ($) - (IFLAG4)*

Subroutine reads the theoretical trajectory data from unit 3 and returns a five point average. For the Interdata program the theoretical data is obtained by a call to subroutine THEOT.

ROBIN

Main program controls all subroutines to obtain total ROBIN output. Specific task is to calculate from actual positional radar data, winds, temperature, pressure and density and then add bias correction terms from theoretical trajectory to produce final data output.

RTDATA**

Subroutine reads one data point from the data buffer produced by reading a unilog tape. The actual read is performed by 1108 library subroutine TAPIO.

SINGLE (IDEG1,NPTFT1,CPl,CV1,CA1)**

Subroutine normalizes the filter coefficients generated by subroutine LEGNDR.
SLIDE (IB,ICNT,$,$) - (IB,ICNT,IFLAG5)*

Subroutine shifts the data in the raw data array by one second. That is two data points (oldest time) are shifted out the top and two new data points (most recent time) are shifted into the bottom. New data is obtained from REAVG.

SLIDET (IB,$) - (IB,IFLAG6)**

Subroutine shifts the data in the theoretical raw data array by one second. Two data points are shifted out the top and new data points are shifted into the bottom. New theoretical data is obtained from REAVGT.

TAB (ESALT,IY)

Subroutine prints a table of amplitude ratios for both density and wind.

TABLE**

Subroutine reads from unit 7 the biases generated by the theoretical trajectory and stores these bias terms in a bias array. In the Interdata program the bias terms are placed directly into the bias array as they are calculated by TROBIN.

THEOT

Subroutine calculates a theoretical trajectory based on the starting point apogee of position, velocity and acceleration and falling of the sphere thru the standard atmosphere with zero winds. The differences between the winds, temper-
atures, etc. found by TROBIN, based on the theoretical trajectory, and data from the standard atmosphere tables produce the bias correction terms. The program for the 1108 system writes the trajectory data to unit 3. The program for the Interdata system returns the next calculated point to the calling program.

TIFALL (HI,TFT,IFT,ZVM,HI2,COLAPS,IPGE,IY)

Subroutine makes an inflation check of a sphere based on the time of fall thru the various layers.

TIFAL2 (HI,TFT,IFT,ZVM,HI2,COLAPS,IPGE,IY)

This subroutine makes the same check as in TIFALL except for a sphere of density .115 or .165.

TROBIN (GS,N2MID,KSW2,KSW,IPGE,IY)

Subroutine uses theoretical trajectory data to compute density, temperature, winds, etc. For the 1108 system, trajectory data is read from unit 3 and the results written to unit 7. For the Interdata system, the data is obtained by successive calls to THEOT and the resulting calculations from TROBIN are put directly into the bias arrays.

WANGLE (THETA,WE,WN)

Subroutine uses east wind (WE), and wind (WN) components and returns polar north direction (THETA).

B. Data Editing Program

The subroutines and programs described here are not incorporated into the Interdata program because they would
need to be changed to accommodate data produced by sphere track from a modified Nike Hercules radar. Since the radar has not been modified to track a sphere, nothing has been done to the editing programs except to better document their purpose.

PMR

Main program copies the data tape onto a temporary file and in the process edits out some systematic dropouts. Also prints a list of the data at 1-second intervals.

AO

Main program calls the editing routines via subroutine MAIN and then calls the main data reduction program ROBIN.

MAIN

Subroutine reads the variables IDCHECK and TSTART, writes them on the output and then calls subroutine MDECK to perform the editing.

MDECK

Subroutine edits data points by discarding those points which do not meet the tolerance requirements. Data thus rejected is replaced by interpolating between good data points.

READST (NT)

Subroutine searches for the first record label on unit NT. Resultant parameters for the record are written to unit 6.
READTT (NT, TI, SI, S2, S3, NA, NCT, NFI, *, *)

Entry point within READST. Reads data from unit 14 until a data point with start time is encountered. Consecutive data points are returned to calling program.

WRITAPE (NA, HT1, HS1, HS2, HS3, M)

Subroutine writes edited data for input to the ROBIN program to unit 15. The individual points are packed into a 96-word record before being written to unit 15.
II. COMPARISON OF INTERDATA AND UNIVAC OUTPUTS

The wind outputs from the Interdata and UNIVAC systems for the benchmark data is compared in Figure 1. This Figure is a graph of the east-west and north-south wind components at 1-kilometer intervals beginning at 98 km and ending at 36 km. Visual comparison of the data shows that the maximum error is 1 m/s in the N-S wind and no difference for the E-W wind.

The pressure, temperature, and density outputs were not compared graphically because of the large dynamic changes that occur in these variables. However, Table 1 contains the actual pressure, temperature and density that was calculated by the two systems at the altitudes indicated.

In comparing pressure, the maximum percentage error was 0.12% which occurred at 81 km. This percentage error is calculated by use of the following equation:

\[
\text{percentage error} = \frac{(\text{ABS(PUNIVAC - PINTERDATA)}) \times 100}{\text{PINTERDATA}}
\]

The nominal error for pressure data calculated for altitudes less than 76 km is less than .06%.

The maximum error in temperature is 1 degree. This 1-degree error occurs because only whole numbers are printed on the output. Internally the error would be approximately the same as for the pressure data. The maximum density percentage error is 0.33% which occurred at 91 km. The nominal error for density data for altitudes less than 83 km is less than .06%.

In all cases the maximum error from Table 1 is due to the fact that output printed did not contain enough significant digits. Using
Table 1. Pressure, Temperature and Density Comparison of UNIVAC and Interdata Outputs Utilizing Benchmark Input Data.

<table>
<thead>
<tr>
<th>Altitude (km)</th>
<th>Interdata Pressure (MB)</th>
<th>UNIVAC Pressure (MB)</th>
<th>Interdata Temperature (Deg K)</th>
<th>UNIVAC Temperature (Deg K)</th>
<th>Interdata Density (G/M^3)</th>
<th>UNIVAC Density (G/M^3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>98</td>
<td>.00038</td>
<td>.00037</td>
<td>184</td>
<td>184</td>
<td>.00072</td>
<td>.00072</td>
</tr>
<tr>
<td>97</td>
<td>.00046</td>
<td>.00046</td>
<td>176</td>
<td>176</td>
<td>.00091</td>
<td>.00091</td>
</tr>
<tr>
<td>96</td>
<td>.00056</td>
<td>.00056</td>
<td>170</td>
<td>171</td>
<td>.00114</td>
<td>.00114</td>
</tr>
<tr>
<td>95</td>
<td>.00068</td>
<td>.00069</td>
<td>165</td>
<td>166</td>
<td>.00144</td>
<td>.00144</td>
</tr>
<tr>
<td>94</td>
<td>.00086</td>
<td>.00084</td>
<td>163</td>
<td>164</td>
<td>.00179</td>
<td>.00179</td>
</tr>
<tr>
<td>93</td>
<td>.00103</td>
<td>.00103</td>
<td>164</td>
<td>165</td>
<td>.00218</td>
<td>.00218</td>
</tr>
<tr>
<td>92</td>
<td>.00126</td>
<td>.00126</td>
<td>170</td>
<td>171</td>
<td>.00257</td>
<td>.00256</td>
</tr>
<tr>
<td>91</td>
<td>.00152</td>
<td>.00152</td>
<td>178</td>
<td>178</td>
<td>.00297</td>
<td>.00296</td>
</tr>
<tr>
<td>90</td>
<td>.00187</td>
<td>.00182</td>
<td>186</td>
<td>186</td>
<td>.00340</td>
<td>.00339</td>
</tr>
<tr>
<td>89</td>
<td>.00216</td>
<td>.00216</td>
<td>192</td>
<td>193</td>
<td>.00390</td>
<td>.00389</td>
</tr>
<tr>
<td>88</td>
<td>.00255</td>
<td>.00255</td>
<td>197</td>
<td>198</td>
<td>.00450</td>
<td>.00449</td>
</tr>
<tr>
<td>87</td>
<td>.00302</td>
<td>.00002</td>
<td>201</td>
<td>201</td>
<td>.00523</td>
<td>.00522</td>
</tr>
<tr>
<td>86</td>
<td>.00356</td>
<td>.00356</td>
<td>202</td>
<td>202</td>
<td>.00613</td>
<td>.00613</td>
</tr>
<tr>
<td>85</td>
<td>.00421</td>
<td>.00421</td>
<td>202</td>
<td>202</td>
<td>.00725</td>
<td>.00724</td>
</tr>
<tr>
<td>84</td>
<td>.00497</td>
<td>.00497</td>
<td>199</td>
<td>199</td>
<td>.00867</td>
<td>.00866</td>
</tr>
<tr>
<td>83</td>
<td>.00588</td>
<td>.00588</td>
<td>198</td>
<td>198</td>
<td>.01033</td>
<td>.01033</td>
</tr>
<tr>
<td>82</td>
<td>.00697</td>
<td>.00696</td>
<td>195</td>
<td>196</td>
<td>.01238</td>
<td>.01238</td>
</tr>
<tr>
<td>81</td>
<td>.00827</td>
<td>.00826</td>
<td>192</td>
<td>192</td>
<td>.01496</td>
<td>.01496</td>
</tr>
<tr>
<td>80</td>
<td>.00964</td>
<td>.00963</td>
<td>189</td>
<td>189</td>
<td>.01806</td>
<td>.01806</td>
</tr>
<tr>
<td>79</td>
<td>.01172</td>
<td>.01171</td>
<td>188</td>
<td>188</td>
<td>.02162</td>
<td>.02162</td>
</tr>
<tr>
<td>78</td>
<td>.01396</td>
<td>.01395</td>
<td>189</td>
<td>189</td>
<td>.02571</td>
<td>.02571</td>
</tr>
<tr>
<td>77</td>
<td>.01662</td>
<td>.01661</td>
<td>190</td>
<td>190</td>
<td>.03039</td>
<td>.03039</td>
</tr>
<tr>
<td>76</td>
<td>.01976</td>
<td>.01976</td>
<td>192</td>
<td>192</td>
<td>.03579</td>
<td>.03579</td>
</tr>
<tr>
<td>75</td>
<td>.02347</td>
<td>.02347</td>
<td>195</td>
<td>194</td>
<td>.04193</td>
<td>.04194</td>
</tr>
<tr>
<td>74</td>
<td>.02782</td>
<td>.02781</td>
<td>198</td>
<td>198</td>
<td>.04887</td>
<td>.04888</td>
</tr>
<tr>
<td>73</td>
<td>.03289</td>
<td>.03290</td>
<td>202</td>
<td>202</td>
<td>.05662</td>
<td>.05664</td>
</tr>
<tr>
<td>72</td>
<td>.03873</td>
<td>.03874</td>
<td>206</td>
<td>205</td>
<td>.06549</td>
<td>.06553</td>
</tr>
<tr>
<td>71</td>
<td>.04549</td>
<td>.04549</td>
<td>210</td>
<td>210</td>
<td>.07517</td>
<td>.07524</td>
</tr>
<tr>
<td>70</td>
<td>.05321</td>
<td>.05323</td>
<td>215</td>
<td>215</td>
<td>.08583</td>
<td>.08594</td>
</tr>
<tr>
<td>69</td>
<td>.06199</td>
<td>.06202</td>
<td>220</td>
<td>220</td>
<td>.09776</td>
<td>.09889</td>
</tr>
<tr>
<td>68</td>
<td>.07020</td>
<td>.07026</td>
<td>225</td>
<td>225</td>
<td>.11130</td>
<td>.11135</td>
</tr>
<tr>
<td>67</td>
<td>.08345</td>
<td>.08350</td>
<td>229</td>
<td>229</td>
<td>.12678</td>
<td>.12675</td>
</tr>
<tr>
<td>66</td>
<td>.09629</td>
<td>.09630</td>
<td>239</td>
<td>239</td>
<td>.14030</td>
<td>.14014</td>
</tr>
<tr>
<td>65</td>
<td>.11062</td>
<td>.11065</td>
<td>241</td>
<td>241</td>
<td>.15944</td>
<td>.15943</td>
</tr>
<tr>
<td>64</td>
<td>.12697</td>
<td>.12702</td>
<td>242</td>
<td>242</td>
<td>.18228</td>
<td>.18224</td>
</tr>
<tr>
<td>63</td>
<td>.14573</td>
<td>.14574</td>
<td>245</td>
<td>245</td>
<td>.20676</td>
<td>.20672</td>
</tr>
<tr>
<td>62</td>
<td>.16677</td>
<td>.16679</td>
<td>250</td>
<td>250</td>
<td>.23183</td>
<td>.23174</td>
</tr>
<tr>
<td>61</td>
<td>.19047</td>
<td>.19048</td>
<td>254</td>
<td>254</td>
<td>.26049</td>
<td>.26045</td>
</tr>
<tr>
<td>60</td>
<td>.21634</td>
<td>.21675</td>
<td>254</td>
<td>255</td>
<td>.29480</td>
<td>.29407</td>
</tr>
<tr>
<td>59</td>
<td>.24526</td>
<td>.24510</td>
<td>259</td>
<td>259</td>
<td>.32907</td>
<td>.32891</td>
</tr>
<tr>
<td>58</td>
<td>.27898</td>
<td>.27881</td>
<td>261</td>
<td>261</td>
<td>.37222</td>
<td>.37204</td>
</tr>
</tbody>
</table>
Table 1. Pressure, Temperature and Density Comparison of UNIVAC and Interdata Outputs Utilizing Benchmark Input Data (Continued).

<table>
<thead>
<tr>
<th>Altitude (km)</th>
<th>Pressure (MB)</th>
<th>Temperature (Deg K)</th>
<th>Density (G/M³)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Interdata</td>
<td>UNIVAC</td>
<td>Interdata</td>
</tr>
<tr>
<td>51</td>
<td>.67338</td>
<td>.67301</td>
<td>263</td>
</tr>
<tr>
<td>50</td>
<td>.76436</td>
<td>.76394</td>
<td>262</td>
</tr>
<tr>
<td>45</td>
<td>1.44703</td>
<td>1.44631</td>
<td>263</td>
</tr>
<tr>
<td>40</td>
<td>2.78673</td>
<td>2.78544</td>
<td>248</td>
</tr>
<tr>
<td>35</td>
<td>5.52564</td>
<td>5.52333</td>
<td>235</td>
</tr>
<tr>
<td>30</td>
<td>11.40067</td>
<td>11.39648</td>
<td>226</td>
</tr>
<tr>
<td>25.140</td>
<td>28.56268</td>
<td>28.56044</td>
<td>140</td>
</tr>
</tbody>
</table>
data where 5 (32 bit floating point accuracy) or more digits are printed, the error was less than .06%. Thus, the internal error is less than .06% in all cases.

The total execution time for calculating and printing the total output was 26 minutes. Part of the total output is included from the initial height to an altitude of 50 km.
| TIME  | ALT. | ENTR. | NHIND | NMIND | SPEED | DIR | PKESS | T | DENSITY | Z VEL | Z ACC | TDEN | CF | DENS | PRFS | TEMP | FH | NN  |
|-------|------|-------|-------|-------|-------|-----|------|---|---------|-------|------|------|----|-----|------|------|------|---|-----|
| ZULTA | MTPM& | M/S  | M/S   | M/S   | DEG   | MR  | K    | G/R3 | M/S    | M/S   | M/S  | /CC | /S  | +++PPKCENT+++ | M/S | M/S  |
| 17    | 48   | 29    | 1/6:35 |       |       |     |     |     | 73     | -9.54 | -9.54 | 300  | 28 | -9.54 | 0.08 | 300  | 28 | -9.54 | 330  | 28 | -9.54 |
| 17    | 48   | 22    | 1/4:922 |      |       |     |     |     | 73     | -9.54 | -9.96 | 309  | 72 | -9.54 | 0.08 | 309  | 72 | -9.54 | 339  | 72 | -9.54 |
| 17    | 48   | 23    | 1/4:374 |      |       |     |     |     | 73     | -9.54 | -9.33 | 327  | 66 | -9.54 | 0.08 | 327  | 66 | -9.54 | 337  | 66 | -9.54 |
| 17    | 48   | 24    | 1/4:019 |      |       |     |     |     | 73     | -9.54 | -9.27 | 337  | 44 | -9.54 | 0.08 | 337  | 44 | -9.54 | 337  | 44 | -9.54 |
| 17    | 48   | 25    | 1/4:682 |      |       |     |     |     | 73     | -9.54 | -9.24 | 346  | 56 | -9.54 | 0.08 | 346  | 56 | -9.54 | 346  | 56 | -9.54 |
| 17    | 48   | 26    | 1/4:839 |      |       |     |     |     | 73     | -9.54 | -9.22 | 355  | 72 | -9.54 | 0.08 | 355  | 72 | -9.54 | 355  | 72 | -9.54 |

**Balloon APOGFE: 138 KM.**
### Ratio of Amplitude of Smoothed Density Wave to Amplitude of Original Wave as a Function of Altitude and Wavelength

(Altitude and Wavelength Measured in Kilometers)

<table>
<thead>
<tr>
<th>Altitude (km)</th>
<th>100</th>
<th>90</th>
<th>80</th>
<th>70</th>
<th>60</th>
<th>50</th>
<th>40</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>W</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.02X</td>
<td>0.57X</td>
<td>0.96X</td>
</tr>
<tr>
<td>V</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>V</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.03X</td>
<td>0.10X</td>
<td>0.40X</td>
<td>0.76X</td>
</tr>
<tr>
<td>L</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>L</td>
<td>0.02X</td>
<td>0.03X</td>
<td>0.06X</td>
<td>0.18X</td>
<td>0.76X</td>
<td>0.98X</td>
<td>0.99X</td>
<td>0.99X</td>
</tr>
<tr>
<td>G</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>G</td>
<td>0.14X</td>
<td>0.17X</td>
<td>0.24X</td>
<td>0.35X</td>
<td>0.97X</td>
<td>0.99X</td>
<td>0.99X</td>
<td>0.99X</td>
</tr>
<tr>
<td>T</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>T</td>
<td>0.58X</td>
<td>0.63X</td>
<td>0.66X</td>
<td>0.90X</td>
<td>0.98X</td>
<td>0.99X</td>
<td>0.99X</td>
<td>0.99X</td>
</tr>
</tbody>
</table>

### Ratio of Amplitude of Smoothed Sinusoidal Wind to Amplitude of Original Wind as a Function of Altitude and Wavelength

(Altitude and Wavelength Measured in Kilometers)

<table>
<thead>
<tr>
<th>Altitude (km)</th>
<th>100</th>
<th>90</th>
<th>80</th>
<th>70</th>
<th>60</th>
<th>50</th>
<th>40</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>W</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.02X</td>
<td>0.57X</td>
<td>0.96X</td>
</tr>
<tr>
<td>V</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>V</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.01X</td>
<td>0.03X</td>
<td>0.10X</td>
<td>0.40X</td>
<td>0.76X</td>
</tr>
<tr>
<td>L</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>L</td>
<td>0.02X</td>
<td>0.03X</td>
<td>0.06X</td>
<td>0.18X</td>
<td>0.76X</td>
<td>0.98X</td>
<td>0.99X</td>
<td>0.99X</td>
</tr>
<tr>
<td>G</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>G</td>
<td>0.14X</td>
<td>0.17X</td>
<td>0.24X</td>
<td>0.35X</td>
<td>0.97X</td>
<td>0.99X</td>
<td>0.99X</td>
<td>0.99X</td>
</tr>
<tr>
<td>T</td>
<td>x</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>T</td>
<td>0.58X</td>
<td>0.63X</td>
<td>0.66X</td>
<td>0.90X</td>
<td>0.98X</td>
<td>0.99X</td>
<td>0.99X</td>
<td>0.99X</td>
</tr>
</tbody>
</table>
III. CONCLUSION

The conversion of the ROBIN program from the UNIVAC 1108 system to the Interdata 7/32 system has been accomplished. The resultant outputs agree to within 1 m/s on winds and .06% on pressure, temperature and density.

The coefficients for Legendre polynomial smoothing can be stored in a data table, thus, saving 32,000 bytes of storage. However, in storing the coefficients in a data table, there must be 4 corrections made in the FORTRAN source code.

The code changes must be made in ROBIN and TROBIN as indicated below.

ROBIN

Line 209
NEND = N2ST + NZ2
Change to
NEND = N2ST + NZ2 - 1

Line 216
NEND = NXY2 + N2ST
Change to
NEND = NXY2 + N2ST - 1

TROBIN

Line 37
NEND = N2ST + NZ2
Change to
NEND = N2ST + NZ2 - 1

Line 44
NEND = NXY2 + N2ST
Change to
NEND = NXY2 + N2ST - 1

Without these changes one non-existent data point is smoothed by a non-existent coefficient. In the present UNIVAC program the non-existent coefficient is set to zero, thus the added smoothed point value is zero and does not change the filtered output. By storing the filter coefficients in a table, the non-existent coefficient is not zero and a large error occurs in the filtered output.
The same smoothing is done in FITON but the correct code exists as indicated.

FITON

Line 16   \[ NEND = N21 + N1ST - 1 \]

Line 28   \[ NEND = NXY1 + N1ST - 1 \]

The program will yield accurate results on the Interdata system.

When data from the Nike radar is obtained, further programs would be required to edit the raw data before running the program.
IV. FLOWCHARTS

UNIVAC 1108 GENERAL DATA FLOW

START

READ FROM UNIT 15
X, Y, Z AND T

HAS THE SPHERE BEGUN FALLING?

COMPUTE THEORETICAL TRAJECTORY DATA POINT

STORE ON UNIT 3 THEORETICAL TRAJECTORY DATA POINT

IS THE ALTITUDE LESS THAN 55 KM?

READ FROM UNIT 3 THEORETICAL TRAJECTORY DATA POINT

COMPUTE THEORETICAL WINDS, TEMPERATURE, AND DENSITY DATA

STORE ON UNIT 7 THEORETICAL WINDS, TEMPERATURE, AND DENSITY DATA

IS THE LAST ALTITUDE REACHED YET?

YES
UNIVAC 1108 GENERAL DATA FLOW

READ FROM UNIT 7
THEORETICAL WINDS,
TEMPERATURE, AND
DENSITY DATA

COMPUTE BIAS VALUES
FROM THE STANDARD
ATMOSPHERE AND
STORE THEM IN THE
BIAS ARRAYS

IS THE LAST ALTITUDE
REACHED YET?

YES

READ FROM UNIT 13
X, Y, Z, and T

WRITE ON OUTPUT
WINDS, TEMPERATURE,
AND DENSITY DATA

CALCULATE WINDS,
TEMPERATURE, AND
DENSITY AND
SUBTRACT THE BIAS

IS ALL THE DATA
PROCESSED?

YES

STOP

NO

NO

IS THE LAST ALTITUDE
REACHED YET?
START

READ FROM UNIT 1
X, Y, Z, AND T

HAS THE SPHERE BEGUN FALLING?

YES

COMPUTE THE THEORETICAL TRAJECTORY POINTS FOR THE FILTER ARRAY

USE THE FILTERED THEORETICAL DATA TO CALCULATE THE THEORETICAL WINDS, TEMPERATURE, AND DENSITY

USING THE THEORETICAL DATA, COMPUTE THE BIAS VALUES FROM THE STANDARD ATMOSPHERE AND STORE THEM IN THE BIAS ARRAY

NO

IS THE ALTITUDE LESS THAN 53 KM?

YES
FLOW

READ FROM UNIT 1
X, Y, Z AND T

CALCULATE WINDS,
TEMPERATURE, AND
DENSITY AND
SUBTRACT THE BIAS

WRITE ON OUTPUT
WINDS, TEMPERATURE,
AND DENSITY DATA

IS ALL
THE DATA
PROCESSED?

YES

STOP

NO
FLOWCHART FOR MAIN PROGRAM ROBIN - 1.1

START
  ↓
READ THE LIMITS FROM UNIT 5
  ↓
CALL SINGLE AND GET LEGENDRE COEFFICIENTS
  ↓
INITIALIZE VALUES AND ESTABLISH MIDPOINTS OF ARRAYS
  ↓
READ UNIT 5 TO GET BALLOON DATA AND INPUT/OUTPUT UNITS USED
  ↓
COMPUTE AND SET CONSTANTS
  ↓
WRITE ALL INFORMATION THAT WE HAVE RECEIVED FROM UNIT 5
  ↓
READ IDD AND COMPARE WITH ID IF THEY ARE NOT EQUAL, STOP. IF THEY ARE, CONTINUE
  ↓
ESTABLISH CONSTANTS AND SET VARIABLES
  ↓
SET IPGE = 6 AND CALCULATE CONVERSION FACTORS
  ↓
CALL CORR,TWICE AND RETURN WITH CORRECTION FACTOR
  ↓
ADD CONVERSION FACTOR AND CORRECTION FACTORS

A1.2
A
1.1

INITIALIZE ALL TIME-OF-FALL TEST VALUES TO ZERO

DETERMINE WHICH MASS BALLOON IS USED OR IF THE BALLOON IS NOT ROCKET

CALL DECAL, RETURNS WHEN 3 DECREASING ALTIMETRIES ARE FOUND

IF IFLAG = 0, CALL CONTROL AND START THE THEORETICAL CALCULATIONS. IF IFLAG # 0, THEN CONTINUE AND SET IFLAG = 1

CALL FITON AND SPLINE TO ESTIMATE THE SMOOTHING POLYNOMIALS

CALL ACCELERATIONS, AND MIDPOINT VELOCITIES AND COORDINATES, AND A NEW ALTITUDE

THROUGH A SERIES OF IF STATEMENTS, MIDPOINT TIME IS RELATED TO TIME OF FALL TESTS.

B
1.3
CALL EITHER TIFALL OR TIFAL2 DEPENDING ON WHAT MASS OF BALLOON IS USED

IF UL IS GREATER THAN THE ALTITUDE BIAS COMPUTED BY THE THEORETICAL SUBROUTINES

CALCULATE THE CORIOLIS ACCELERATION

IF ISW = 1

CALL ATMOS, COMPUTE REYNOLDS’S AND MACH NUMBER AND CALL DRAG TO GET A COEFFICIENT OF DRAG

COMPUTE DENSITY GRADIENT AND IF IT EXCEEDS A NOMINAL VALUE COMPUTE IF IT HAS COLLAPSED

COMPUTE BUOYANCY AND WINDS

COMPUTE VELOCITY AND DENSITY

RECOMPUTE MACH AND REYNOLDS NUMBERS AND CALL DRAG AND SET ICU = 1

GO INTO A LOOP INVOLVING EXPERT CHANGING DENSITY AND DRAG UNTIL CURRENT DENSITY TIMES THE PREVIOUS DENSITY DIVIDED BY CURRENT DENSITY IS LESS THAN 1/100TH

SECOND RETURN

SECOND RETURN
FLORCHART FOR MAIN PROGRAM ROBIN - 1.4

C 4

INITIALIZE BIAS VALUES TO ZERO

IF ALTITUDE IS LESS THAN 60 KM

IF ALTITUDE IS LESS THAN 60 KM

COMPUTE RAHUS DEPENDING ON MACH NUMBER AND BIAS MACH NUMBER

COMPUTE TIME, DENSITY, AND WIND BIAS INCREMENTS

COMPUTE TIME, DENSITY, AND WIND CORRECTION FACTORS, COMPUTE WIND AND CALL WANGL WHICH RETURNS A DIRECTION

COMPUTE VELOCITIES

CALCULATE TIME AND SHIFT VALUES OF ALTITUDE, REYNOLD'S NUMBER, AND PRESSURE TO PREVIOUS VALUE STORAGE

IF INTERP = 2 CALL INTERP FOR INTERPOLATED DATA AND 1000 INTER PRINTOUT. IF NOT, CONTINUE

IF IPOR LESS THAN 56, CONTINUE. IF NOT SET IPOR = 3

E 3
FLOWCHART FOR MAIN PROGRAM ROUTIN - 1.5

IF HI GREATER THAN HIASHI(1)

YES

IF INTERP = 2

WRITE TIME (HR, MIN, SEC) ALTITUDE 22, 23

NO

WRITE

NO

IF INTERP = 2

SET ISW = 2

IF INTERP = 0, WRITE ALT, I, RHO, MUS, PINB, WE, WI ON UNIT 10 IF NOT, CONTINUE

NO

INCREMENT IPGE AND WRITE COMPUTED DATA

INCREMENT ICD

IF ICD IS LESS THAN OR EQUAL TO 5, CONTINUE IF NOT, SET ICD = 1 AND ISW = 1

SHIFT MIDPOINT POSITIONS AND VELOCITIES ONE PLACE TO THE RIGHT

CALL FITON AND SLIDE TO GET NEW VALUES

SECOND RETURN

THIRD RETURN

39
FLOWCHART FOR WICK PROPELLOR ROBIN - 1.6

IF HI LESS THAN 60 KM
AND TIME AT MIDPOINT
IS GREATER THAN
TIME OF FALL TEST 6
PLUS 50 SECONDS

IF Z VELOCITY AT
MIDPOINT ≠ ZVC

RECALCULATE CONVERSION FACTORS
AND CALL CORRS FOR NEW CORRECTION
FACTORS

IF NZI = 51

CHANGE LIMITS
AND RECALL SINGLE

IF 2 VELOCITY AT
MIDPOINT # ZVC

Determine what
ALTITUDE BALLOON
IS AT AND WHICH 1FT
TO SEND TO TIPRSS

READ IDD

IF ID = IDD

WRITE "NORMAL END
OF ROBIN"
WRITE ON UNIT 10
ALTITUDE, TIME, DENSITY,
PRESSURE WINDS, AND
COLLAPSE ALTITUDE

REWIND UNIT 10
STOP
FLOWCHART FOR SUBROUTINE ATMOS - 2.1

START

FIND TABLE INDEX FOR HEIGHT

FROM INDEX VALUE, INTERPOLATE VALUES IN STANDARD ATMOSPHERE TABLES, TO GET TEMPERATURE AND DENSITY FOR HEIGHT

RETURN
FLOWCHART FOR SUBROUTINE BLKDAT - 3.1

DATA TABLES
FOR ALL COMMON
DATA VARIABLES
START

SET POSITION EQUAL TO NZ1 MIDPOINT FOR X,Y, AND Z

FIND LINEAR LEGENDRE VELOCITIES USING P21

DEFINE CONSTANTS (GSRG, AM, RAD, CZB, SZB, OMEGA, CPH, SPH, CPHSZB, CPHCZB) AND INITIALIZE WX AND WZ EQUAL TO ZERO

CALL THEOT AND RETURN WITH X,Y,Z, AND T ON UNIT 3 (ALL THEORETICAL)

SET CONSAV(I) EQUAL TO GAR(I) FOR I=1,1010

CALL TROBIN AND RETURN WITH THE DENSITY, TEMP, WINDS PRESSURE, ETC. FOR THE THEORETICAL TRAJECTORY COMPUTER BY THEOT. PUT ON UNIT 7

CALL TABLE WHICH TAKES THE ABOVE DATA ON UNIT 7 AND COMPUTES THE BIASES TO BE USED IN THE ACTUAL DESCENT

SET GAR(I) EQUAL TO CONSAV(I) FOR I=1,1010

RETURN
FLOWCHART FOR SUBROUTINE CORRS - 3.1

START

ESTABLISH DENOMINATOR TERM D
D = \sum_{i=1}^{N} [A(i)]^2

DO LOOP STARTING POINT
J=1, M AND K=1, M

YES
IF J EQUALS K

NO

ESTABLISH IR AS THE INTEGER ABSOLUTE VALUE OF (IS*(J-K))
SINCE IS ALWAYS Equals 2, \nIR Equals THE INTEGER AABSOLUTE VALUE TO TWICE THE DIFFERENCE.
ALSO ESTABLISH THAT IN IS N MINUS IR

YES
IF IN IS LESS THAN OR EQUAL TO ZERO

NO

CALCULATE RMSR USING THE TIME FORMULA RMSR = \sum_{i=1}^{N} [A(i)]A(i+IR)
FOR I=1,1,M AND THEN DIVIDE BY D

CALCULATE C2 = \sum_{j=1}^{M} [6(j)B(j)*RMSR]
END DO LOOP

RETURN
FLOWCHART FOR SUBROUTINE DECAL - 8.1

START

ESTABLISH R = 6371229.315
AND ALT = -R

CALL READPT WHICH RETURNS
AN X, Y, Z, AND TIME POINT

RETURN 2
RETURN 3

CHANGE X, Y, AND Z COORDINATES:
FROM METERS TO FEET

NEGATIVE
 ZERO

SET ICNT EQUAL TO ZERO
AND SET ALT
EQUAL TO ALT 1

NEGATIVE
 ZERO

CALCULATE ALTITUDE, ALT1,
AND SUBTRACT FROM ALT

POSITIVE

POSITIVE

INCREMENT ICNT
AND SUBTRACT
FROM 5

POSITIVE

IF FT-T

NEGATIVE
 ZERO

RETURN

CONVERT THE TIME
TO INTEGER VALUE

CALL READPT AND GET
X,Y,Z, AND TIME

RETURN 2
RETURN 3
FLOWCHART FOR SUBROUTINE DLV - 7.1

START

\[\downarrow\]

COMPUTE POSITIONAL CHANGES

\[\downarrow\]

COMPUTE DENSITY AND GRAVITATIONAL CHANGES

\[\downarrow\]

COMPUTE WIND CHANGES

\[\downarrow\]

RETURN
START

MACH NUMBER BOUNDED IN TABLE?

NO

YES

IS REYNOLDS NUMBER BOUNDED?

NO

PRINT NO DRAG COEFFICIENT

YES

DO TWO DIMENSIONAL INTERPOLATION TO FIND DRAG COEFFICIENT

RETURN 2

RETURN
FLOWCHART FOR MAIN PROGRAM DRIVE - 9.1

START

READ DATA FROM FIELD TAPE

END OF FILE?

YES

STOP

NO

CONVERT DATA TO RANGE AZIMUTH AND ELEVATION

CALCULATE CURVED EARTH X,Y, AND Z COORDINATES

WRITE X,Y,Z, AND TIME TO TAPE UNIT 14
FLOWCHART FOR SUBROUTINE DRVT - 10.1

START

CALCULATE ALTITUDE, H, GIVEN FROM THEOT AND ALSO COMPUTE THE VARIABLES TO BE USED IN CALCULATING THE DENSITY

IF K EQUAL 2

YES

PLACE HI BETWEEN ALT(L) AND ALT(L-1)

NO

TAKE PREVIOUS DRAGS AND MODIFY

CALCULATE VELOCITY, DENSITY, TEMPERATURE, REYNOLDS NUMBER, AND MACH NUMBER FROM THE THEORETICAL DATA GIVEN BY THEOT

CALL DRAG AND WRITE HI AND CALCULATE A DRAG

RETURN

49
FLOWCHART FOR SUBROUTINE FITON - 11.1

START

INITIALIZE VARIABLES TO ZERO

CALCULATE SMOOTHING VELOCITIES, 3 CUBIC AND ONE SET OF LINEAR VELOCITIES

ESTABLISH MIDPOINT VALUES AND SMOOTHING VARIABLES

RETURN
START

CHOOSE INCREMENTS OF EITHER 200 TO 1000 METERS IF INTERP EQUALS 1 OR 2 RESPECTIVELY

NO

IF HI IS GREATER THAN STALT

NEGATIVE

ZERO

IF INTALT

POSITIVE

CHANGE ALTITUDE TO EVEN 200 or 1000 METER ALTITUDES

ESTABLISH VALUES OF SA1 to SA19

RETURN

NEGATIVE

IF INTALT MINUS THE INTEGRAL VALUE OF HI

ZERO

INTERPOLATE NEW VALUES OF SA1 to SA19 USING DELALT JUST ESTABLISHED

YES

IF IPGE LESS THAN 56

NO

SET IPGE = 3 AND WRITE TITLE BLOCK ON NEW PAGE

WRITE DATA

INCREMENT IPGE, DECREMENT INTALT
FLOWCHART FOR SUBROUTINE LEGNUR - 13.1

START

COMPUTE COEFFICIENTS FOR ORDER (10 MAX) AND NUMBER (100 MAX)

COMPUTE COEFFICIENTS FOR VELOCITY (FIRST DERIVATIVE) CALCULATIONS

COMPUTE COEFFICIENTS FOR ACCELERATION (SECOND DERIVATIVE) CALCULATIONS

RETURN
START

IF ICNT ≤ 24

CALL READ READS
THE DEVICE WITH ROBIN INPUT DATA ON IT
I=1,96

IF END OF FILE

WRITE 0(1) ON UNIT 6
RETURN 8

IF D(2) = -999.3

WRITE 0(1) ON UNIT 6
RETURN 8

RETURN 7

IF D(1) = CHR

RETURN 7

ICNT = 1

SET I = 4(ICNT) + 1
ESTABLISH S1,S2,S3,7 IN TERMS OF D

INCREMENT ICNT

RETURN
FLOWCHART FOR SUBROUTINE REAVG - 17.1

START

INITIALIZE VARIABLES TO ZERO

CALL READPT FIVE TIMES AND SUM X'S, Y'S, Z'S, AND T'S

RETURN 2
RETURN 3

DIVIDE TIME BY 5, X, Y, AND Z BY 16.404

RETURN
FLOWCHART FOR SUBROUTINE REAVG - 18.1

START

INITIALIZE VARIABLES TO ZERO

READ UNIT 3 FOR 5 VALUES OF T, X, Y, Z AND SUM THE X'S, Y'S, Z'S, AND THE T'S.

RETURN 1

DIVIDE THE SUMMATIONS BY 5 TO GET AVERAGES

RETURN
START

READ FIELD TAPE RECORD

END OF FILE?

YES

NO

GET SR IN FEET, AZ IN RADIANS, AND EL IN RADIANS

COMPUTE CURVED EARTH ALTITUDE AND X AND Y COMPONENTS

RETURN

RETURN
FLOWCHART FOR SUBROUTINE SINGLE - 19.1

START

CALL LEGNDR TO GET POLYNOMIAL COEFFICIENTS FOR DEGREE (10 MAX) AND NUMBER (100 MAX)

NORMALIZE COEFFICIENTS

RETURN
FLOWCHART FOR SUBROUTINE SLIDE - 20.1

START

SHIFT ALL VALUES 2 PLACES, TIME, X, Y, AND Z

CALL REAVG TO GET 2 NEW VALUES

RETURN

59
START

MOVE THEORETICAL DATA IN ARRAY BY ONE POINT

RETURN
FLOWCHART FOR SUBROUTINE TAB - 22.1

START

WRITE MAIN HEADING FOR AMPLITUDE RATIO

ALT > 75km? 

YES

PRINT DATA FROM TABLES FOR THIS ALTITUDE

NO

ALT > 90km? 

YES

PRINT DATA FROM TABLES FOR THIS ALTITUDE

NO

ALT > 100km? 

YES

PRINT DATA FROM TABLES FOR THIS ALTITUDE

NO

ALT > 110km? 

YES

PRINT DATA FROM TABLES FOR THIS ALTITUDE

NO

PRINT DATA FROM TABLES FOR THIS ALTITUDE

RETURN
START

INITIALIZE J AND I

READ UNIT 7 TO GET HI, VMC, WH, VR, RHO, T

IF END OF FILE

YES RETURN

NO

IF ACT(I) ≤ HI

NO

DECREMENT I

UNTIL ALT(I) ≤ HI

YES

COMPUTE CORRECTIONS FOR THEORETICAL TEMPERATURE AND DENSITY

COMPUTE OF BIASES

INCREMENT J

NO

IF J = 600

YES RETURN

NO
START

INITIALIZE VALUES TO ZERO AND SET TIME INCREMENTS TO .1 SEC

ESTABLISH A POSITION AND CALL DRMT WHICH DETERMINES IF A DRAG VALUE IS GIVEN BY CALLING DRAG

NUMERICALLY INTEGRATE TO GET A NEW POSITION

INCREMENT TIME BY .1 SEC AND WRITE TIME, X(1), Y(1), Z(1) ON UNIT 3

COMPUTE A THEORETICAL ALTITUDE, TALT

IF TALT IS GREATER THAN 55000 THEN

NC

SET T = -999.9 AND WRITE T, X(1), Y(1), Z(1) ON UNIT 3 THEN END OF FILE

RETURN
FLOPCRT FOR SUBROUTINE TIFAIL - 25.1

Tifail is dependent on IFT as to where in the subroutine it will return to. The first statement is a computed Go TO statement with eleven possible choices.

If IFT = 1, return immediately.

IFT = 2

START

NO

IF ZVM(2) IS LESS THAN ZVM(1)

YES

IF HI IS GREATER THAN OR EQUAL TO 77900 METERS

NO

IFT = 2

YES

IF ZVM(1) IS GREATER THAN OR EQUAL TO -200

YES

SET IFT = J

RETURN

SET IFT = 2 AND IFT = 1, WRITE "APOGEE NOT KNOWN" AND INCREMENT IPEE

RETURN

64
FLOWSHART FOR SUBROUTINE TIFALL - 25.2

IFT = 3

START

IF ZVM(2) IS GREATER THAN OR EQUAL TO ZVM(1)

YES

RETURN

NO

IF ZVM IS LESS THAN OR EQUAL TO 200 METERS

RETURN

YES

SET IFT = 5 AND CALCULATE ALTITUDES TO BE USED TO WRITE APOGEE

ADD 3 TO IPGE AND SET IFT = 6

RETURN

IFT = 4

START

A 25.2

SET IFT = 4

IF ZVM(1) IS LESS THAN - 150

NO

SET IFT = 11

YES

IF ZVM(1) IS LESS THAN - 210

YES

SET IFT = 5 AND IFT = 1 WRITE "APOGEE NOT KNOW" AND INCREMENT IPGE

RETURN

NO

RETURN

RETURN
FLOWCHART FOR SUBROUTINE TIFALL

**START**

IF ATFT IS LESS THAN OR EQUAL TO ZERO

SET JFT EQUAL TO CURRENT JFT PLUS ONE

IF TPT, TIME OF FALL TESTS, IS BETWEEN 2 NUMBERS DIFFERENT FOR EACH JFT

WRITE "COLLAPSE BETWEEN 3 CM INCREMENTS OF ALT.

IF COLAPS = 0

SET COLAPS = THE LOWER OF THE TWO ALTITUDES ABOVE (IN METERS)

INCREMENT IPGE

RETURN

WRITE "BALLOON STILL INFLATED AT," SOME PRESET KILOMETER ALTITUDE, ALSO INCREMENT IPGE

RETURN

RETURN

66
IFT = 6

START

SAME AS IFT = 5
BUT INSTEAD OF
SETTING JFT,
TTEST IS SET
ACCORDING TO
ALTITUDE PARAMETERS

TTEST IS THEN
USED AS A MIDPOINT
FOR THE TIME OF
FALL TESTS

THE REMAINDER OF
IFT = 6 IS STRUCTURED
LIKE IFT = 5

RETURN

IFT = 7, IFT = 8, AND IFT = 10 ARE
SIMILAR TO IFT = 5.

IFT = 9 AND IFT = 11 ARE STRUCTURED
THE SAME AS IFT = 3, BUT THEY
HAVE NUMERICAL DIFFERENCES
FLOWCHART FOR SUBROUTINE TIFALZ - 26.1

START

SAME AS FOR TIFALL
(EXCEPT USE .165m SPHERE)

RETURN

68
INITIALIZE ISW, ICO, W1, and NPRT

CALL REAVGT AND GET 51 VALUES FOR TIME, X, Y, AND Z.

CALL FITON AND SLIDET TO GET SMOOTHING FACTOR FOR ALL DATA POINTS 51 TIMES

CALCULATE 4 ACCELERATIONS

CALCULATE VELOCITIES AND POSITIONS AND ALTITUDE

IF 2 ACCELERATION IS LESS THAN -8 METERS PER SECOND^2

NO

COMPUTE THE CORIOLIS ACCELERATIONS

IF ISW = 1

YES

CALL ATMOS, COMPUTE REYNOLD'S NUMBER, AND MACH NUMBER AND CALL DRAG WHICH RETURNS THE DRAG COEFFICIENT

NO

REARRANGE DRAG COEF.
FLOWCHART FOR SUBROUTINE TROI3IN - 47.1

开始

1. 计算浮力和风
2. 计算速度和密度
3. 重新计算马赫数和雷诺数
4. 计算升力和阻力并设置ICD = 1
5. 如果当前密度减去前一次密度小于1/100，则
   - 否
   - 是，计算加速度
8. 如果加速度小于8米/秒²，
   - 否
   - 是，输出：高度，马赫数，风速，密度，时间
9. 设置ISW = 2并增加ICD
10. 如果ICD小于5，
    - 是
    - 否
11. 设置ICD = 1和ISW = 1

结束
FLOWCHART FOR SUBROUTINE TROBIN - 27.3

C

27.3

SET NPRT AND K

SHIFT MID VELOCITY COORDINATES ONE SPACE TO THE RIGHT

CALL FITON AND SLIDET TO GET SMOOTHING FACTOR FOR ALL DATA POINTS

A

27.1
FLOWCHART FOR SUBROUTINE WANGL - 28.1

START

COMPUTE WIND VECTOR
DIRECTION ANGLE

RETURN
FLOWCHART FOR MAIN PROGRAM PSR - E1.1

START

WRITE ON OUTPUT
COLON HEADINGS

WRITE ON 14
FIRST RECORD
LABEL

READ
INPUT
TAPE ID

INCREMENT
RECORD
COUNTER

WRITE ON OUTPUT
TAPE ID

READ
ONE DATA
RECORD

EOF?

YES

B
1.2

NO

INCREMENT
RECORD
COUNTER

A
1.2

C
1.1
A 2.2
READ NEXT DATA RECORD
EUF?
YES
NO
INCREMENT RECORD COUNTER
SAVE SECOND SET OF T, Y, AND Z AS THE FIRST
CONVERT THE FIRST Y AND Z, SECOND X, FROM METERS TO FEET
WRITE ON OUTPUT FIRST T, Y, AND Z WITH THE SECOND X
WRITE ON 14 TAPE ID, FIRST T, Y, AND Z WITH THE SECOND X
IS THE SECOND TIME ZERO?
YES
STOP
WRITE ON 14 LAST RECORD LABEL
MARK AND REWIND THE OUTPUT FILE
WRITE TO UNIT 27 THE ID RECORD
WRITE ON OUTPUT NUMBER OF RECORDS COPIED
FLOWCHART FOR MAIN PROGRAM A® - E2.1

START
CALL MAIN
CALL ROBIN
WRITE ON 27 LAST RECORD LABEL
WRITE ON 4 CONTENTS OF UNIT 27
STOP
FLOWCHART FOR SUBROUTINE MAIN - E3.1

START

READ FROM CASE
IDCHEK
IDSTART

WRITE ON OUTPUT
IDCHEK
IDSTART

CALL MDECK

RETURN
FLOWCHART FOR SUBROUTINE MDECK - E4.1

START

INITIALIZE

READ FROM CARDS
EDITING PARAMETERS

WRITE ON OUTPUT
EDITING PARAMETERS

READ FROM CARDS
FLIGHT ID

WRITE ON OUTPUT
FLIGHT ID

CALL READST

INCREMENT FILE COUNTER

WRITE ON 15
FLIGHT ID

SECOND RETURN

CALL READST

THIRD RETURN

INCREMENT NUMBER
OF RECORDS READ

IS THE ALTITUDE
LESS THAN 150000
FEET?

YES

NO

4.2

K

4.1

3

4.5
FLOWCHART FOR SUBROUTINE MDECK - E4.3

IS THE TIME DIFFERENCE BETWEEN POINTS GREATER THAN .3 SECONDS?

YES

NO

INCREMENT COUNTER FOR BAD POINTS IN SEQUENCE

IS THE TIME DIFFERENCE BETWEEN POINTS LESS THAN .1 SECOND?

YES

NO

RESET COUNTER FOR BAD POINTS IN SEQUENCE

ARE X, Y, AND Z VALUES ALL WITHIN TOLERANCE?

YES

NO

LET PREVIOUS TIME BE THE PRESENT TIME

E 4.4

D 4.3

B 4.2

B 4.2

C 4.2

B 4.2

79
CALL U-1%TE

EXACT

DIFFERENCE

BETWEEN

PC

WRITE ON OUTPUT

END OF FLIGHT INDICATOR

INCREMENT DISCARDED POINT COUNTER

BEGIN A NEW FLIGHT

WRITE ON OUTPUT

END OF FLIGHT INDICATOR

ARE X, Y, AND Z VALUES ALL WITHIN AN APPROPRIATELY SCALED TOLERANCE?

GENERATE BY INTERPOLATION THE NEXT TENTH SECOND POINT

IS THE TIME DIFFERENCE GREATER THAN 2 SECONDS?

IS THE GENERATED POINT STILL BEFORE THE TIME OF THE NEXT DATA POINT?
FLOWCHART FOR SUBROUTINE MDECK - E4.5

K
G
SECOND RETURN
SECOND RETURN
THIRD RETURN
GET NEXT INCREMENT COUNT
POINT VIA CALL READD
INCREMENT NUMBER OF RECORDS READ
RESET COUNTER FOR BAD POINTS IN SEQUENCE
C
J
MARK 15
WRITE ON OUTPUT NUMBER OF GENERATED POINTS
NUMBER OF DISCARDED POINTS
NUMBER OF RECORDS WRITTEN
REWIND 15
RETURN

H
INCREMENT COUNTER FOR GENERATED POINTS
WRITE ON 15 GENERATED POINT
INCREMENT NUMBER OF RECORDS WRITTEN
WILL THE NEXT GENERATED POINT BE WITHIN .001 SECONDS OF THE PRESENT DATA POINT?
YES
NO
INCREMENT TIME BY .1 SECOND
I
START

INCREMENT COUNTER FOR RECORDS READ

IS THIS THE SECOND RECORD? NO

YES

WRITE ON OUTPUT RECORD COUNTER INPUT UNIT OUTPUT UNIT

READ FROM 14 DATA POINT

WAS LAST RECORD INDICATOR READ? NO

YES

WRITE ON IS DUMMY DATA RECORD

WRITE ON OUTPUT DUMMY DATA RECORD

WRITE ON OUTPUT EOF HAS OCCURRED

MARK 15

A 6.2

NORMALIZE TIME OF DATA POINT TO TSTART

B 6.2
FLOWCHART FOR SUBROUTINE READT - E6.2

A

6.2

IS THIS THE SECOND RECORD?

YES

WRITE ON OUTPUT
TSTART

INTERCHANGE THE
ORDER OF THE
DATA VALUES

NO

WRITE ON OUTPUT

B

6.2

FILE COUNT
NUMBER OF EOF
ENCOUNTERED
RECORD COUNT

ARE ALL FILES
PROCESSED?

NO

RETURN 9

RETURN

YES

RETURN 10
V. PROGRAM LISTING FOR INTERDATA COMPUTER

```
* $HSSM

ROBIN  PROG RIGID SPHERE TRAJ  30 NOV 79 RO1  S REM3:ROBIN FOR
CROSS
NOR3
NLSTC
NLIST

* FORT

REAL NA
DOUBLE PRECISION DX1, DY1, DZ1, DRG, DHMSL, DHI
INTEGER*2 IFLAG, IFLAG2, IFLAG3, IFLAG4, TP, CN, IMX, IMY, IP1
INTEGER*2 IDEGK1, IDEGX2, IDEG2, IDEG72, IDEG71, N2HID
INTEGER*2 NXY3, NXY2, NX1MID, NX2MID, NX1MID, N2ST
INTEGER*2 JX, JN, MIDMAX, KMAX1, KMAX2, N1ST, NXY1, N1Z
INTEGER*2 BC, IBCC, IBCH, IFLAG6, ID IN, 1, IOO, INTERP, IGN, NPERT
INTEGER*2 IC, IPGF, IBAL, IFT, ISNM, IMU, IHR, JS1T, K, KSH, KSN2, NEND

COMMON /TF/TFL, TF2, TFL2, TFH, TF7, TF8, TFS, TFG, TFG2
COMMON /D,X1, Y1, Z1, Y2, Z2, 2X, 2Y, 2Z, X1, X2, X3, Y1, Y2, Y3, Z1, Z2
COMMON /CON1, CON2, CON3, CON4, GS, IDEG2, IDEG1, IDEG72, IDEG71
COMMON /BIAS/ IMX, BIAS1(600), BIAS2(600), BIAS3(600)
COMMON /BIAS/(JMX, BIAS4(JMX), BIAS5(JMX), BIAS6(JMX)
COMMON /CONST/ RG, ALA, HMSL, AMS, ZB, DIA, VR, AM, BSRG, AB
COMMON /EXTRA/ NXY2, NX1MID, NX2MID, NX1MID, N2ST
COMMON /DEF/ PX1(31), PX2(35), PX3(21), PX4(31)
COMMON /DOR2/ TOSL, TOCL, TOCLS2, TOCLS2
COMMON /CON2/ AMK, ALPHA, RU, RAD
COMMON /INT/IH, HI, AMK, UF, WN, WT, THETA, PICOR, RHOCOR
COMMON /TCOR/, CD, RE, VMH, VWH, VWH, VWH, VWH, VWH, VWH, VWH, VWH
COMMON /TIME/(100), X(100), Y(100), Z(100), THD(50), XHD(50), YHD(50), ZHD(50)
COMMON /1H/, IH1, IH2, IH3, IH4, IH5, IH6, IH7, IH8, IH9, IH10
COMMON /D1/ HI, NXY1, NXY2, NXY3, NXY4

DIMENSION CP1(100), CA1(100), ANAME(20)

DATA ANAME/40H?32 TEST OF ROBIN BENCHMARK DATA.

102 FORMAT(11I8, 2I8), F6. 2, 21X, F7. 1, F6. 1, F7. 4, F8. 5, 13)
324 FORMAT(1I8, 20A4)
325 FORMAT(6SH0) ID RD WR ALA GS RG HML, ZB AMS
* DRA INTH, IN, 212, F6. 2, F11. 6, F10. 1, F7. 1, F6. 1, F7. 4, F8
* S, 13)
329 FORMAT(20A4)
400 FORMAT(1H, 65X, 27H***R. M. S. NOISE ERROR IN***/2X. 119HTIME ALT.
* EWIN R. NOISE SPEED DIR PRESS T DENSITY Z VEL Z ACC TDE
*N CF DENS PRES TEMP EN NH/3X. 120H20ULU METERS M/S
* M/S M/S DEG MR K GR/M3 M/S M/S2 /CC
* M/S *****PERCENT***** M/S M/S )

86
```
401. FORMAT(1H3, 95X, 27H***R. M. S. NOISE ERROR IN***/3X, 11SHTIME ALT.
   * EWIN0 NWIND SPEFD DIR PRESS T DENSITY Z VEL. Z ACC TDE
   * N CF DENS PRFS TEMP EW NH/3X, 120HZULU METERS M/S
   * M/S M/S DEG MB K GR/M3 M/S M/S2 /CC
   *  /S  *****PERCENT***** M/S M/S )
403 FORMAT(1X, I2, 2(1H, , I2), 17, 2F6. 1, F7. 2, 2(14, F9. 5), F8. 2, F6. 2, 2(1X, E9
   * . 4), 3F6. 1, 2F5. 1)

C

479 FORMAT(47H1JNIVERSITY OF DAYTON ROBIN PROGRAM - SEPT 1977 //)
481 N21=19
482 CALL LINEAR(NZ1)
483 IRC=2
484 IBCA=2
485 IBCM=2
486 IFLAGR=0
487 ISW=1
488 RE=0.
489 PICOR=0.
490 RHOHGS=0.
491 TCOR=0.
492 TND2=0.
493 PI=0.
494 ZVC=0.
495 COLAPS=0. 0
496 NZ1MID=10
497 N1ST=17
498 N1MID=26
499 N2MID=18
500 N2MID=11
501 KMAX2=35
502 KMAX4=51
503 MIDMAX=26
504 N2MID=18
505 N2ST=8
506 C2=1.
507 NXY1=51
508 NXY2=35
509 N22=21
510 IDEGX1=3
511 IDEGX2=3
512 IDEGZ1=1
513 IDEGZ2=3
514 1011 FORMAT(1H0, 12X, 30HXY-VFL. XY-ACC Z-VEL. Z-ACC /
515 A11H = PTS. FIT. 418/  
516 R11H DEGREE FIT. 418)

C

THE FOLLOWING READS ARE REPLACED BY EQUALITY STATEMENTS

87
C ? RFAU (5, 32, 9) NAME
102 C READ (5, 102) ID, IX, IY, ALA, HMLS, ZB, AMS, DIA, INTERP
103 ID = 2377
104 ALA = 52.46
105 HMLS = 1220
106 ZB = 4
107 AMS = 1649
108 DIA = 1
109 INTERP = 1
110 RAD = 57.2957795
111 FG = 67.7939.14
112 C GS = -9.011; 1; 1; 0.299; 52; 0.2985E-10 + (SIN(ALA/RAD) + 2) - 5.9E-8 + (SIN(2 * ALA/RAD))
113 C GS = -9.011; 1; 1; 0.299; 52; 0.2985E-10 + (SIN(ALA/RAD) + 2) - 5.9E-8 + (SIN(2 * ALA/RAD))
114 C X = RAD - 3
115 WRITE (15, 124) RS AMF
116 WRITE (19, 1611) XNY1, XNY2, NZ1, NZ2, IDEG1, IDEG2, IDEG3, IDEG4, IDEG5
117 WRITE (19, 1305) ID, IX, IY, ALA, GS, RG, HMLS, ZB, AMS, DIA, INTERP
118 3001 FORMAT (415)
119 ICNT = 25
120 C 220 READ (5, 102) ID
121 222 ID = 2377
122 IF (IDP = ID) GO TO 226, 222
123 222 JSIT = 1
124 IF (JSTT Eq. 0) STOP
125 GO TO 226
126 226 VB = (0.527 * 45) + DIA + DIA + DIA
127 AB = (0.299 * 89998) + DIA + DIA
128 ESHIT = 0.6
129 ICD = 1
130 AMI = 0.008367
131 AMFR = -0.0001923
132 RU = 8.34137
133 PTN = 0.0790F + 16
134 NA = 6.0 + 169E + 23
135 RSNR = 0.30132
136 WZ1 = 6.0
137 NPRT = 0
138 PIMB = 0.0
139 CZE = COS (ZB/RAD)
140 SZE = SIN (ZB/RAD)
141 TOL5 = 0.000168 + SIN(ALA/RAD)
142 TOL6 = 0.000168 + COS(ALA/RAD)
143 TOL54 = TOL5 + SZE
144 TOCL52 = TOL5 + CZE
145 IFGE = 12
146 DEL = 25
147 ANI = XNY1
148 CON1 = 12. / (DEL + AN1 + (AN1 * AN1 = 1))
149 CON2 = CON1 + 72; *(3. * AN1 + AN1 - 7) ** 2 / (DEL + AN1 + (AN1 + AN1 - 1))
150 j*(AN1 + AN1 - 4) * (AN1 * AN1 - 9)
AN1=N21
CONZ1=12. / (DEL*AN1*(AN1*AN1-1.))
CONZ2=CONZ1+7. *(3. *AN1*AN1-1. )**2 / (DEL*AN1*(AN1*AN1-1.))
1*(AN1*AN1-4.) *(AN1*AN1-9.)
AN1-NXY2
DEL=.1.
CONX3=12. / (DEL*AN1*(AN1*AN1-1.))
CONX4=CONX3+7. *(3. *AN1*AN1-1. )**2 / (DEL*AN1*(AN1*AN1-1.))
1*(AN1*AN1-4.) *(AN1*AN1-9.)
AN1-NZ3
CONZ3=12. / (DEL*AN1*(AN1*AN1-1.))
CONZ4=CONZ3+7. *(3. *AN1*AN1-1. )**2 / (DEL*AN1*(AN1*AN1-1.))
1*(AN1*AN1-4.) *(AN1*AN1-9.)
CALL CORRS(NXY1, PXY1, NXY2, PXY2, RMSN, 2)
CALL CORRS(NZ1, PZ1, NZ2, PZ2, RMSD, 2)
CONX3=CONX3+RMSN
CONX4=CONX4+RMSN
CONZ3=CONZ3+RMSD
CONZ4=CONZ4+RMSD
VP=0.0
TFT:0.0
TFT1:0.0
TFT2:0.0
TFT3:0.0
TFT4:0.0
TFT5:0.0
TFT6:0.0
TFT7:0.0
TFT8:0.0
TFT9:0.0
TFT10:0.0
IF(ABS(DIA1-1.) LT. 0.02. AND. ABS(AMS-. 115). LT. 0.01.) GO TO 300
IF(IBAL=1.) GO TO 230
300 CALL DECALT(ICNT, IFLAG1)
IF(IFLAG1=301, 35, 90)
301 WRITE(IY, 400)
IFGE=IFGE+3
INTALT=0
DO 52 J=1, KMAX1
52 CALL REAVG(ICNT, IFLAG3)
IF(IFLAG3=1003, 40, 90)
1003 IF(KMAX1. NE. NXY1) GO TO 3
251 Y2=YVM(N2MID)
252 2X=ZXM(N2MID)
253 2=-ZVM(N2MID)
254 X1=XMID(N2MID)
255 Y1=YMID(N2MID)
256 Z1=ZMID(N2MID)
257 TIM=TMID(N2MID)
258 HI=Z1+(X1*X1+Y1*Y1)/(2.*RG)+HMSL
259 IH=HI
260 ENOHI=ZMID(1)+(XMID(1)**2+YMID(1)**2)/(2.*RG)+HMSL
261 OPHROR=1+HI/RG
262 WTHR=GS/(RG+OPHOR*OPHOR*OPHOR)
263 IF(TFT6.EQ.0.0.AND. HI.LT.60000.) TFT6=TIM
264 IF(TFT1.NF.0.0.OR. HI.GT.80000.) GO TO 500
265 TFT1=TIM
266 500 IF(TFT2.NE.0.0.OR. HI.GT.70000.) GO TO 501
267 TFT2=TIM
268 TFT=TFT2-TFT1
269 501 IF(TFT3.NF.0.0.OR. HI.GT.60000.) GO TO 502
270 TFT3=TIM
271 TFT=TFT3-TFT2
272 502 IF(TFT4.NE.0.0.OR. HI.GT.55000.) GO TO 503
273 TFT4=TIM
274 503 IF(TFT5.NE.0.0.OR. HI.GT.50000.) GO TO 505
275 TFT5=TIM
276 505 IF(TFT7.NE.0.0.OR. HI.GT.45000.) GO TO 506
277 TFT7=TIM
278 506 IF(TFT8.NE.0.0.OR. HI.GT.40000.) GO TO 507
279 TFT8=TIM
280 507 IF(TFT9.NE.0.0.OR. HI.GT.35000.) GO TO 508
281 TFT9=TIM
282 508 IF(TFT10.NE.0.0.OR. HI.GT.30000.) GO TO 509
283 TFT10=TIM
284 509 IF(TFT11.NE.0.0.OR. HI.GT.25000.) GO TO 510
285 TFT11=TIM
286 510 IF(HI.GT.BIASHI(1)) GO TO 39
287 ENOLO=ZMID(KMAX2)+(XMID(KMAX2)**2+YMID(KMAX2)**2)/(2.*RG)+HMSL
288 CORX=TOCLSZ+Z2X+TOSL+Y2
289 CORY=TOCLSZ+Z2X+TOSL*X2
290 CORZ=-TOCLSZ+X2-TOCLSZ*Y2
291 GO TO (19,38), ISW

91
19 CALL HMSH(RH00,RH00)
20 IF (ABS(Z2+SORTT(1, 0)+X3+X3+Y3+Y3)/(Z3-WTHR*(Z1+RG))**2)*2>
30 AND (RH00+T*(T+116.4)/(0.000001458+SORT(T+T+T)))
40 AND (Z2+SORTT(268, 16/T))/340.29205
50 CALL (NFSI(CD,AMC,RF,IFLAG))
60 IF (Z2+SORTT(1, 0)+X3+X3+Y3+Y3)/(Z3-WTHR*(Z1+RG))**2 >
70 AND (RH00+T*(T+116.4)/(0.000001458+SORT(T+T+T)))
80 AND (Z2+SORTT(268, 16/T))/340.29205
90 CALL (NFSI(CD,AMC,RF,IFLAG))
100 IF (Z2+SORTT(1, 0)+X3+X3+Y3+Y3)/(Z3-WTHR*(Z1+RG))**2 >
110 AND (RH00+T*(T+116.4)/(0.000001458+SORT(T+T+T)))
120 AND (Z2+SORTT(268, 16/T))/340.29205
130 CALL (NFSI(CD,AMC,RF,IFLAG))
IF (IFLHG < 512, 2000, 512)
512
ICD=1
513
IF (NPRTEQ.0) GO TO 26
514
CD=CD0+CD-CD0)/3.
515
IF (ARS (RHO-RHOO)/RHO, LT. 0.003333) GO TO 28
516
IF (NPRTEQ.5) GO TO 27
517
26
RHOO=RHO
518
NPRT=NPRT+1
519
GO TO 20
520
27
RHOO (RHO+RHOO)/2. 0
521
CD=(CD+CD0)/2. 0
522
NPRT=0
523
GO TO 20
524
28
RHOO=RHO
525
NPRT=0
526
RHOO=RHO
527
TBS=0 0
528
RBS=0 0
529
WXBS=0 0
530
WYBS=0 0
531
IF (IBC1GE JMK) GO TO 45
532
IF (HI LT 60000, ) GO TO 45
533
IF (AMC LE AMC, AND. AMC, GE . 85, AND. AMC, LE 1. 3) GO TO 460
534
46
IF (BIASHI (IBCA), LE HI) GO TO 47
535
IBCA=IBCA+1
536
GO TO 46
537
47
IM1=IBCA
538
IBCA=IBCA-1
539
RATIO (HI-BIASHI (IM1))/(BIASHI (IBC)-BIASHI (IM1))
540
GO TO 475
541
460
IF (BIASHMC (IBCM), LE AMC) GO TO 476
542
IBCM-IBCM+1
543
GO TO 460
544
476
IM1=IBCM
545
IBCM=IBCM-1
546
RATIO (AMC-BIASHMC (IM1))/(BIASHMC (IBC)-BIASHMC (IM1))
547
475
TBS-BIASHI (IM1) (BIASHI (IBC)-BIASHI (IM1))*RATIO
548
RBS-BIASHR (IM1) (BIASHR (IBC)-BIASHR (IM1))*RATIO
549
WXBS-BIASHWX (IM1) (BIASHWX (IBC)-BIASHWX (IM1))*RATIO
550
WYBS-BIASHY (IM1) (BIASHY (IBC)-BIASHY (IM1))*RATIO
551
AMC=AMC
552
45
TCOR=T-TBS
553
RHOCOR=RHO-RBS
554
WXCOR=NXWXBS
555
WYCOR=WY-WYBS
556
PICOR=TCOR+RHO+RHOCOR/AMK
557
WE=WXCOR+ZB+WYCOR+C2B
558
WN=WXCOR+C2B-WYCOR+ZB
559
WT=SORT(WF**2+WN**2)
560
CALL WANG1 (THFTR, WE, WN)
THE DESCRIPTION OF THE ROBIN PROGRAM AND ITS CONVERSION TO THE --ETC(U)

MAY 80 M D MERRILL, D ELWELL

ERADCOM/ASL-CR-80-0115-1 NL
IHETA=THETA
RHOMGS=RHOCOR*1000.
CALL DEV(VRHO, VWX, VWY, VPR, RHOMGS)
VP=SQRT(VPR)/(PI*999.831)
VT=SQRT(VP**2+VRHO**2)*100.0
VP=VP*100.0
VRHO=VRHO*100.
VN=SQRT((VWX*CZR)**2+(VWY*SZR)**2)
VE=SQRT((VWX*CZR)**2+(VWY*SZR)**2)
VRHO=VRHO
VWY=VW
TND=NI+PICOR/(RSTAR*TCOR)+1.0E-6
IF(PLT=TIM
MIN=ITIM/60
ISEC=ITIM -60+MIN
IHR=MIN/60
MIN=MIN-60+IHR
PIM=0I+PICOR
CFREG=PICOR*7*80064E+5
TND=RHOMGS*RTN
IT=TCOR+5
IRE=RE
PI1=P1
HI=HI
ALT=IH
IF(INTERP.GE.1) CALL INTER(INTERP, Z2, Z3, IY, IPGF, BIAISHI(1))
IF(IPGF.LT.56) GO TO 33
IPGF=3
WRITE(IY, 401)
33 CONTINUE
IF(HI.GT.BIAISHI(1)) GO TO 43
IF(INTERP.EQ.0) WRITE(4, 3000) ALT, T, RHOMGS, PIMB, WE, WN
3000 FORMAT(' ', 6E10.4)
ISW=2
IF(INTERP.EQ.2) GO TO 2000
IPGF=IPGF+1
WRITE(IY, 403) IHR, MIN, ISEC, IH, WE, WN, WT, IHETA, PIMB, IT, RHOMGS, Z2, Z3,
*TND2, CFREG, VRHO, VP, VT, VWX, VWY
2000 GO TO 2000
43 CONTINUE
IF(INTERP.EQ.2) GO TO 2050
IF(IPGF.EQ.1)
WRITE(IY, 434) IHR, MIN, ISEC, IH, Z2, Z3
434 FORMAT(1X, 12, 2(IH, 12), 17, 45X, F8.2, F6.2)
455 GO TO 2050
2000 ICD=ICD+1
IF(IPGF.EQ.5) GO TO 2050
ICD=1
ISW=1
2050 CONTINUE
NPRT=0
32  K=KMAX2-1
   DO 31  I=1,K
      IP1=I+1
55   XMD(I)=XMD(IP1)
56   YMD(I)=YMD(IP1)
57   ZMD(I)=ZMD(IP1)
58   TMID(I)=TMID(IP1)
59   XMID(I)=XMID(IP1)
60   YMID(I)=YMID(IP1)
61   ZMID(I)=ZMID(IP1)
62   ZVM(I)=ZVM(IP1)
63   GO TO (29,30),KSN2
   29  N=KMAX2
      CALL FITON
      CALL SLIDE(2,ICNT,IFLAG5)
      IF(IFLAG5)240,40,90
   30  N=KMAX2
      CALL FITON2
      CALL SLIDE(2,ICNT,IFLAG5)
      IF(IFLAG5)240,40,90
   240  IF(HI.LT.60000. .AND. TIM. GT. TFT6+50.) GO TO 250
   GO TO 260
   250  IF(NZ1.EQ.51 ) GO TO 260
      NZ1=NZ1+4
      TFT6=TFT6+50.
      NZ1MID=NZ1/2+1
      NLST=ABS(NX1MID-NZ1MID)+1
      ZVC=ZVM(KMAX2)
      CALL LINEAR(NZ1)
      GO TO 11
      260  IF(ZVM(N2MID). NE. ZVC) GO TO 11
   48   AN1=NZ1
   49   DEL=. 25
   50   CON2=CON1+7.*((AN1*AN1-9.)**2/(AN1*AN1-1.))
   51   CON3=CON3+RMSD
   52   CON4=CON4+RMSD
   53   CALL CORRS(NZ1,PZ1,NZ2,PZ2,RMSD,2)
   54   CONZ3=CONZ3+RMSD
   55   CONZ4=CONZ4+RMSD
   56   GO TO 11
   40  REAI)<IX) IDD
   40  IF(ID-IDD) 90,36,90
   36  ICNT=25
   35  INALT=0
   38  IF(HI.LT.50000.) GO TO 301
   39  IF(HI.LT.60000.) GO TO 1501
   500  IF(HI.LT.70000.) GO TO 1502
501 IF(H1.LT.80000.) GO TO 1503
502 GO TO 301
503 1501 IF(I)=1
504 GO TO 301
505 1502 IF(I)=10
506 TH1=999.
507 GO TO 301
508 1507 IF(I).EQ.0
509 TH2=999.
510 GO TO 361
511 C 35 READ(IN), IDD
512 35 IF(ID-IDD) 1500,37,1500
513 37 ION=0
514 GO TO 220
515 88 WRITE(IY,89)
516 89 FORMAT('***** DENSITY IS NON-POSITIVE -- EXECUTION HALTED *****')
517 90 ALT=-999.
518 WRITE(IY,91)
519 91 FORMAT('1/10X,19HNORMAL END OF ROBIN)
520 WRITE (4,3000) ALT,T,PHOMGS,PMB,WE,WN
521 CALL TARCHES(T,IV)
522 ITIM=-999
523 STOP
524 1500 STOP
525 525 $ASSM
526 LIST
527 $FORT
528 END
SUBROUTINE RTMOS(HIT, RHOO)

INTEGER*2 TP(302), DN(302), J, TNPW, I, KH, JOFS(5)

REAL HSTS(5), HSTP(5)

C RFVISFD TABLE FOR 1976 STANDARD ATMOSPHERE

C 50KM - 100KM IN 0.5KM STEPS

C 100KM - 141KM IN 1.0KM STEPS

DATA JOFS, HSTP/261, 161, 71, 51, 1, 1000, 500, 200, 100, 200./
DATA HSTS/100000, 50000, 32000, 30000, 20000./
DATA (TP(1), I=1,74)/ 21665, 21678,
X 21698, 21718, 21738, 21758, 21778, 21797,
21
X 21817, 21837, 21857, 21877, 21897, 21917,
22
X 21936, 21956, 21976, 21996, 22016, 22036,
23
X 22056, 22075, 22095, 22115, 22135, 22155,
24
X 22175, 22194, 22214, 22234, 22254, 22274,
25
X 22294, 22313, 22333, 22353, 22373, 22393,
26
X 22413, 22432, 22452, 22472, 22492, 22512,
27
X 22532, 22551, 22571, 22591, 22611, 22631,
28
X 22651, 22666, 22686, 22706, 22726, 22746,
29
X 22766, 22789, 22819, 22849, 22879, 22909,
30
X 22929, 22959, 22989, 23019, 23049, 23079,
31
X 23099, 23129, 23160, 23190, 23220, 23250,
32
X 23280, 23310, 23340, 23370, 23400, 23430,
33
X 23460, 23490, 23520, 23550, 23580, 23610,
34
X 23640, 23670, 23700, 23730, 23760, 23790,
35
X 23820, 23850, 23880, 23910, 23940, 23970,
36
X 24000, 24030, 24060, 24090, 24120, 24150,
37
X 24180, 24210, 24240, 24270, 24300, 24330,
38
X 24360, 24390, 24420, 24450, 24480, 24510,
39
X 24540, 24570, 24600, 24630, 24660, 24690,
40
X 24720, 24750, 24780, 24810, 24840, 24870,
41
X 24900, 24930, 24960, 24990, 25020, 25050,
42
X 25080, 25110, 25140, 25170, 25200, 25230,
43
X 25260, 25290, 25320, 25350, 25380, 25410,
44
X 25440, 25470, 25500, 25530, 25560, 25590,
45
X 25620, 25650, 25680, 25710, 25740, 25770,
46
X 25800, 25830, 25860, 25890, 25920, 25950,
47
X 25980, 26010, 26040, 26070, 26100, 26130,
48
DATA (TP(1), I=161,229)/ 27065, 27065, 27065, 27065,
49
X 26990, 26765, 26629, 26490, 26352, 26215,
50
X 26077, 25940, 25802, 25664, 25527, 25389,
<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>51</td>
<td>X</td>
<td>25252, 25114, 24977, 24840, 24702, 24565,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52</td>
<td>X</td>
<td>24425, 24298, 24153, 24015, 23878, 23741,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53</td>
<td>X</td>
<td>23664, 23466, 23329, 23192, 23055, 22918,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54</td>
<td>X</td>
<td>22781, 22644, 22506, 22369, 22232, 22095,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>X</td>
<td>21958, 21821, 21685, 21548, 21412, 21276,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56</td>
<td>X</td>
<td>21231, 21133, 21035, 20938, 20840, 20742,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57</td>
<td>X</td>
<td>20645, 20547, 20449, 20352, 20254, 20156,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58</td>
<td>X</td>
<td>20059, 19961, 19864, 19766, 19669, 19571,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59</td>
<td>X</td>
<td>19474, 19376, 19279, 19182, 19084, 18987,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60</td>
<td>X</td>
<td>DATA (TP(I), I=230,302)/ 18987,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>X</td>
<td>18889, 18792, 18687, 18582, 18477, 18372,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62</td>
<td>X</td>
<td>18667, 18562, 18457, 18352, 18247, 18142,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63</td>
<td>X</td>
<td>18036, 17931, 17826, 17721, 17616, 17511,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>X</td>
<td>17406, 17301, 17196, 17091, 16986, 16881,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>65</td>
<td>X</td>
<td>16776, 16671, 16566, 16461, 16356, 16251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>66</td>
<td>X</td>
<td>16146, 16041, 15936, 15831, 15726, 15621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>67</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>68</td>
<td>X</td>
<td>15306, 15201, 15196, 15091, 14986, 14881,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>69</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>71</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>72</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>73</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>74</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>75</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>76</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>78</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>79</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>80</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>81</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>82</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>83</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>84</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>85</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>86</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>87</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>88</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>89</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>91</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>92</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>93</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>95</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>96</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>97</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td>X</td>
<td>15516, 15411, 15306, 15201, 15196, 15091,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>99</td>
<td>X</td>
<td>14776, 14671, 14566, 14461, 14356, 14251,</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>X</td>
<td>14146, 14041, 13936, 13831, 13726, 13621,</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
101 X 1252, 1222, 1192, 1162, 1134, 1106.
102 X 1079, 1053/
103 DATA (DN(I), I=161, 229) / 1027, 9658, 9069, 8530,
104 X 8056, 7606, 7179, 6749, 6390, 6026,
105 X 5681, 5354, 5044, 4751, 4474, 4211,
106 X 3963, 3728, 3505, 3295, 3097, 2909,
107 X 2732, 2565, 2407, 2258, 2118, 1985,
108 X 1868, 1743, 1632, 1528, 1430, 1337,
109 X 1250, 1168, 1092, 1019, 9517, 8880,
110 X 8283, 7722, 7197, 6704, 6237, 5795,
111 X 5382, 4997, 4639, 4304, 3992, 3702,
112 X 4331, 4045, 3764, 3505, 3295, 3097,
113 X 2732, 2565, 2407, 2258, 2118, 1985,
114 X 1868, 1743, 1632, 1528, 1430, 1337,
115 X 1250, 1168, 1092, 1019, 9517, 8880,
116 X 8283, 7722, 7197, 6704, 6237, 5795,
117 X 5382, 4997, 4639, 4304, 3992, 3702,
118 X 4331, 4045, 3764, 3505, 3295, 3097,
119 X 2732, 2565, 2407, 2258, 2118, 1985,
120 X 1868, 1743, 1632, 1528, 1430, 1337,
121 X 1250, 1168, 1092, 1019, 9517, 8880,
122 X 8283, 7722, 7197, 6704, 6237, 5795,
123 X 5382, 4997, 4639, 4304, 3992, 3702,
124 X 4331, 4045, 3764, 3505, 3295, 3097,
125 X 2732, 2565, 2407, 2258, 2118, 1985,
126 X 1868, 1743, 1632, 1528, 1430, 1337,
127 X 1250, 1168, 1092, 1019, 9517, 8880,
128 DO 10 KH=1,5
129 IF(HI. GE. HSTS(KH)) GO TO 20
130 CONTINUE
131 KH=KH-1
132 20 DLH=HSTP(KH)
133 IDLH=DLH
134 IHI=(INT(HI/DLH))*IDLH
135 J=(IHI-HSTS(KH))/DLH+JOS(KH)
136 30 IF(J.LE.276) GO TO 40
137 TP2=TP(J+1)/10.
138 IF(J.EQ.277) GO TO 45
139 TP1=TP(J)/10.
140 GO TO 50
141 40 TP2=TP(J+1)/100.
142 45 TP1=TP(J)/100.
143 50 TNPW=-5
144 IF(J.GT.88) TNPW=-6
145 IF(J.GT.161) TNPW=-7
146 IF(J.GT.198) TNPW=-8
147 IF(J.GT.228) TNPW=-9
148 IF(J.GT.254) TNPW=-10
149 IF(J.GT.270) TNPW=-11
150 IF(J.GT.288) TNPW=-12
151    DN1=DN(J)*(10.**TNPW)
152    IF((J.EQ.80).OR.(J.EQ.161).OR.(J.EQ.198))GO TO 60
153    IF((J.EQ.278).OR.(J.EQ.254).OR.(J.EQ.270))GO TO 60
154    IF((J.EQ.288))GO TO 60
155    GO TO 76
156    60    TNPW=TNPW-1
157    76    DN2=DN(J+1)*(10.**TNPW)
158    RATIO=(HI-IHI)/DLH
159    T=TP1+RATIO*(TP2-TP1)
160    RHO0=DN1*((DN2/DN1)**RATIO)
161    100    CONTINUE
162    102    RETURN
163    END
<table>
<thead>
<tr>
<th>#ASSM</th>
<th>SCRAT</th>
<th>BLKDATA</th>
<th>PROG</th>
<th>INITIALIZE COEFS 14 NOV 79 R01 S REM3:BLKDATA.FOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>CROSS</td>
<td>NORX3</td>
<td>NL.STC</td>
<td>NL.IST</td>
<td>$FORT BLOCK DATA COMMON /COFF/PXY1(51), PXY2(35), PZ2(21), PZ1(51)</td>
</tr>
<tr>
<td>DATA PXV1/</td>
<td>.0049275309,</td>
<td>X</td>
<td>.0032924339,</td>
<td>.0018334202,</td>
</tr>
<tr>
<td>X</td>
<td>.0015605104,</td>
<td>.0023856888,</td>
<td>-.0030772295,</td>
<td>-.003338298,</td>
</tr>
<tr>
<td>X</td>
<td>.0040657036,</td>
<td>-.0043801926,</td>
<td>-.0045863556,</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>.0046976998,</td>
<td>-.0046110079,</td>
<td>-.0044846876,</td>
<td>-.0042128451,</td>
</tr>
<tr>
<td>X</td>
<td>.0033110482,</td>
<td>-.0035505621,</td>
<td>-.0031387159,</td>
<td>-.0026828498,</td>
</tr>
<tr>
<td>X</td>
<td>-.0021963803,</td>
<td>-.0016684036,</td>
<td>-.0011244968,</td>
<td>-.000569165,</td>
</tr>
<tr>
<td>X</td>
<td>.0000000000,</td>
<td>.000569165,</td>
<td>.0011244968,</td>
<td>.0016684036,</td>
</tr>
<tr>
<td>X</td>
<td>-.0021963803,</td>
<td>-.0026628498,</td>
<td>-.0031387159,</td>
<td>-.003595621,</td>
</tr>
<tr>
<td>X</td>
<td>.0039110482,</td>
<td>-.0042128451,</td>
<td>-.0044846876,</td>
<td>-.0046110079,</td>
</tr>
<tr>
<td>X</td>
<td>.0046926998,</td>
<td>-.0046863556,</td>
<td>-.0045846291,</td>
<td>-.0043001926,</td>
</tr>
<tr>
<td>X</td>
<td>.0048057036,</td>
<td>-.0036338298,</td>
<td>-.0030772295,</td>
<td>-.0023856888,</td>
</tr>
<tr>
<td>X</td>
<td>-.0015605104,</td>
<td>-.0005431455,</td>
<td>-.0005857183,</td>
<td>-.0018334202,</td>
</tr>
<tr>
<td>X</td>
<td>-.0032924339,</td>
<td>-.0049275309/</td>
<td>DATA PXV2/</td>
<td>.0097315972,</td>
</tr>
<tr>
<td>X</td>
<td>-.0024974160,</td>
<td>-.0051528104,</td>
<td>-.0071784742,</td>
<td>-.0086228512,</td>
</tr>
<tr>
<td>X</td>
<td>.0095343813,</td>
<td>-.009615045,</td>
<td>-.0095629898,</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>-.0088208355,</td>
<td>-.0077947341,</td>
<td>-.0065264329,</td>
<td>-.0050643682,</td>
</tr>
<tr>
<td>X</td>
<td>.0034569805,</td>
<td>-.0017527987,</td>
<td>-.0000000000,</td>
<td>-.0017527987,</td>
</tr>
<tr>
<td>X</td>
<td>.0034569805,</td>
<td>-.0050643682,</td>
<td>-.0065264329,</td>
<td>-.0077947341,</td>
</tr>
<tr>
<td>X</td>
<td>.0088208355,</td>
<td>-.0095626898,</td>
<td>-.0095626898,</td>
<td>-.0099615045,</td>
</tr>
<tr>
<td>X</td>
<td>.0095343813,</td>
<td>-.0086228512,</td>
<td>-.0071784742,</td>
<td>-.0051528104,</td>
</tr>
<tr>
<td>X</td>
<td>.0024974160,</td>
<td>-.0008361526,</td>
<td>-.0049963241,</td>
<td>-.0097315972/</td>
</tr>
<tr>
<td>X</td>
<td>-.0024974160,</td>
<td>-.0097315972,</td>
<td>-.0049963241,</td>
<td>-.0008361526,</td>
</tr>
<tr>
<td>X</td>
<td>.0097315972,</td>
<td>.0097315972,</td>
<td>.0097315972,</td>
<td></td>
</tr>
<tr>
<td>END</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
$HSIM

SCAT

CONTRL PROG CONTROLS FLOW 30 NOV 79 RO1 S REM3:CONTRL.FOR

CROSS

NORX3

NLSTC

NLST

#FORT

SUBROUTINE CONTRL(GS, N2MID, KSW2, KSW, IPGE, IY)

COMMON /CONST/ RPI, PHI, HMSL, AMS, ZB, DIA, VB, AB, GSRG, AB

COMMON /COE/ PXV1(51), PXV2(51), PZ2(21), PZ1(51)

COMMON /THEO/ ZT(3), RZ(3), DZ, XT(3), RX(3), DX, YT(3)

COMMON /CORIS/ OMEGA, CPH, CPHCZB, CPHSZB, CZB, SPH, SZB

COMMON /TIME/ TIME(100), X(100), Y(100), Z(100), TMID(50), XMID(50), YMID(50)

A ZMID(50), XMID(50), YMID(50), ZVM(50), XM(50), JN, N, MIDMAX, KMAX1, KMAX2

B CJ, NI, NXY1, NXY2, ZVM(50)

C

INTEGER*2 IT, JN, MIDMAX, KMAX1, KMAX2, NI, NXY1, NXY2

C

C

DIMENSION IGAR(806), ICOMSV(806)

C

EQUIVALENCE (TIME(I), IGAR(I))

C

MG=NI*2+1

C

XT(I)=X(MG)

C

YT(I)=Y(MG)

D

ZT(I)=Z(MG)

C

XT(2)=0

C

YT(2)=0

C

ZT(2)=0

C

DO 1 I=1, MG

C

XT(2)=XT(2)+PZ1(1)*X(I)*2.0

C

YT(2)=YT(2)+PZ1(1)*Y(I)*2.0

C

ZT(2)=ZT(2)+PZ1(1)*Z(I)*2.0

C

GSRG=GS/RG

C

AM=2.0*AB/AMS

C

RAD=9.29527795

C

CZH=COS(ZB/RAD)

C

SZB=SIN(ZB/RAD)

C

OMEGA=0.0001458

C

CPH=COS(PHI/RAD)

C

SPH=SIN(PHI/RAD)

C

CPHCZB=CPH*CZH

C

CPHSGZB=CPH+SZB

C

MNX=0.0

C

MW=0.0

C

DO 10 I=1, 806

C

10 ICOMSV(I)=IGAR(I)

C

CALL TROBIN(GS, N2MID, KSW2, KSW, IPGE, IY)

C

DO 20 I=1, 806

C

20 IGAR(I)=ICOMSV(I)

C

RETURN

C

END
SUBROUTINE CORRS(N, A, M, B, C2, IS)

DIMENSION A(1), B(1)
INTEGER*2 I, J, K, IN, IR, IW, M, N

D=0.0
DO 1 I=1, N
1 D=D+A(I)*A(I)

C2=0.0
DO 10 J=1, M
10 DO 20 K=1, M
20 IF(J.EQ.K) GO TO 50

IR=IAABS(IS*(J-K))
IN=N-IR
RMSR=0.0
IF(IN.LE.0) GOTO 50

DO 10 I=1, IN
10 IW=I+IR

RMSR=RMSR+A(I)*A(IW)

RMSR=RMSR/D
C2=C2+B(J)*B(K)*RMSR

50 CONTINUE
RETURN

END
$ASSM

SCRAT

DECaLT PROG SPHERF DROPPING 14 NOV 79 R01 S R&M3:DECaLT FOR
CROSS
NORM3
NLSTC
NLIST

$FOR1

$PROCEDURE DECaLT(JCNT, IFLAG1)
COMM.IP TIME(100), X(100), Y(100), Z(100), TMID(50), XMID(50), YMID(50)
R(J1, NJST, NXY1, NZ1, ZXM(50))
INTEGER IBND, ICNT, JCNT, IFLAG1
INTEGER IX, J, NJMAX, KMAX1, KMAX2, NJST, NXY1, NZ1
IFLAG1=-1
R=6371229.316
SI=0
SI=0
SI=0
C DATA READ 51=x,S2=y,S3=z
READ(1, 20, END=8) T, S1, S2, S3
FORMAT(4(E18.7)
DIFS=((S1-511)*(S1+S11)+(S2-521)*(S2+S21))
DLTH=S3-S3-DIFS/(R*6.561666)
IF(DLTH) 2, 2, 3
2 ICNT=0
GTO 4
3 ICNT=ICNT+1
IF(G-ICNT) 5, 5, 4
S1=S1
S2=S2
S3=S3
GO TO 1
5 IBND=7
TM=T+10.0
IF(IX(TH)+IBND)/10.
READ(1, 20, END=8) T, S1, S2, S3
IF(FT-T) 7, 7, 6
7 RETURN
8 IFLAG1=0
RETURN
$ASSM
LIST
$FOR1
END

104
$ASSM

SCRAIT

$FORT

SUBROUTINE DEV(VRHO, VWX, VWY, VP, RH000)
COMMON TIME(100), X(100), Y(100), Z(100), TMID(50), XMID(50), YMID(50),
A ZMID(50), XMID(50), YMID(50), ZMID(50), IX, J, N, MIDMAX, KMAX1, KMAX2,
B C1, N1ST, NXY1, NZ1, ZXM(50)
COMMON /D/ X1, X2, X3, Y1, Y2, Y3, Z1, Z2, Z3, Z4, CONX1, CONX2, CONX3,
A CONX4, CONZ1, CONZ2, CONZ3, CONZ4, GS, IDEGX1, IDEGX2, IDEGX1, IDEGZ2, IDEGZ1,
B, N2MID

INTEGER*2 IX, J, N, MIDMAX, KMAX1, KMAX2, N1ST, NXY1, NZ1
INTEGER*2 IV, IDEGX1, IDEGX2, IDEGX1, IDEGZ2, N2MID

DATA IV/*0/.

VARR=36
VARE=(.15)*(15)+.96383E-6
VARA=VARE
TH050=X1+X1+Y1+Y1
THRS0=TH050+Z1+Z1
VARX=X1+X1*X1+VARX/THRS0+Y1+Y1+VARA
VARX=X1+Y1+VARX/THRS0+Y1+1+Z1+VARA/THRS0+X1+Z1+VARA
VARZ=Z1+Z1+VARZ/THRS0+TH050*VARA

IF(IDEGX1.NE.1) GO TO 310
VARXV=CONX1*VARX
VARYV=CONX1*VARY
VARXV=CONX1*VARZ
GO TO 320

310 VARXV=CONX2*VARX
VARYV=CONX2*VARY
VARXV=CONX2*VARZ
GO TO 340

320 IF(IDEGX2.NE.1) GO TO 330
VARX=CONX3*VARX
VARY=CONX3*VARY
VARXZ=CONX3*VARXZ
GO TO 340

330 VARX=CONX4*VARX
VARY=CONX4*VARY
VARXZ=CONX4*VARXZ
GO TO 340

340 IF(IDEGZ1.NE.1) GO TO 350
VARZV=SQRT(CONZ1*VARZ)
GO TO 360

350 VARZV=SQRT(CONZ2*VARZ)

360 IF(IDEGZ2.NE.1) GO TO 370
VARZ2=SQRT(CONZ3*VARZ2*2)
GO TO 380

370 VARZ2=SQRT(CONZ4*VARZ2*2)

105
51     VRHO=(2.*VARZV/(Z2)**2+(VARZA/(Z3-GS))**2
52         IF(IV.EQ.0.1) GO TO 390
53         VP=GS**2/4.0*VRHO*(ZMID(N2MID)-ZMID(N2MID+1))**2*RHOO**2
54         IV=1
55         GO TO 400
56     390     VP=VP+GS**2/4.0*VRHO*(ZMID(N2MID-1)-ZMID(N2MID+1))**2*RHOO**2
57     400     VRHO=SQR(T(VRHO))
58     400     VMY=SQR(2/(Z2X/(Z3X-GS))**2*VARXV+(X3/(Z3X-GS))**2*VARXV+
59             1*(Z3X-ZG)**2)**2*VARXZA)
60     400     VMY=SQR(2/(Z2X/(Z3X-GS))**2*VARYV+(Y3/(Z3X-GS))**2*VARXV+
61             1*(Z3X-ZG)**2)**2*VARXZA)
62     400     RETURN
63     400     $ASSM
64     400     LIST
65     400     $FORT
66     400     END
SUBROUTINE DRAGT(CD, ANM, RF, IFLAG)

INTEGER*2 AR1A(24, 22), AR3(17, 31), AR3(16, 15)
INTEGER*2 RLNR(40), AMACH(53)
INTEGER*2 AR1A(24, 17), AR1B(24, 7), AR1C(24, 7)
INTEGER*2 AR2(17, 8), AR2B(17, 7), AR2C(17, 8), AR2D(17, 8)
INTEGER*2 AR1B(17, 1), J, J1, I, JJ, NMA, IFLAG, ILJ, IL

EQUIVALENCE (AR1(1, 1), AR1A(1, 1)), (AR1(1, 9), AR1B(1, 1)),
X (AR1(1, 16), AR1C(1, 1)), (AR2(1, 1), AR2A(1, 1)),
X (AR2(1, 9), AR2B(1, 1)), (AR2(1, 16), AR2C(1, 1)),
X (AR2(1, 24), AR2D(1, 1))

DATA AR1A/
X167.3, 1604, 1535, 1491, 1449, 1394, 1342, 1289, 1258, 1199, 1171, 1142,
X1132, 1104, 1075, 1069, 1055, 1038, 0, 0, 0, 0, 0, 0
X1683, 1612, 1541, 1491, 1452, 1392, 1347, 1292, 1252, 1200, 1171, 1144,
X1134, 1106, 1086, 1069, 1055, 1041, 0, 0, 0, 0, 0, 0
X1697, 1622, 1547, 1495, 1458, 1400, 1352, 1298, 1254, 1202, 1171, 1147,
X1136, 1108, 1088, 1068, 1055, 1043, 0, 0, 0, 0, 0, 0
X1720, 1648, 1553, 1493, 1463, 1410, 1356, 1302, 1259, 1207, 1175, 1150,
X1138, 1110, 1082, 1055, 1055, 1055, 0, 0, 0, 0, 0, 0
X1749, 1658, 1567, 1519, 1473, 1426, 1366, 1310, 1266, 1212, 1184, 1157,
X1141, 1111, 1088, 1058, 1058, 1051, 0, 0, 0, 0, 0, 0
X1780, 1680, 1580, 1533, 1485, 1438, 1375, 1319, 1275, 1220, 1192, 1163,
X1149, 1122, 1035, 1076, 1062, 1055, 0, 0, 0, 0, 0, 0
X1809, 1783, 1597, 1548, 1498, 1444, 1390, 1338, 1287, 1230, 1182, 1173,
X1158, 1130, 1101, 1088, 1066, 1066, 0, 0, 0, 0, 0, 0
X1841, 1728, 1615, 1564, 1512, 1456, 1400, 1341, 1296, 1242, 1214, 1186,
X1170, 1140, 1089, 1097, 1084, 1072, 0, 0, 0, 0, 0, 0/

DATA AR1B/
X1870, 1775, 1636, 1582, 1526, 1471, 1414, 1355, 1310, 1257, 1227, 1196,
X1179, 1146, 1113, 1100, 1089, 1075, 1063, 1050, 0, 0, 0, 0
X1903, 1788, 1657, 1601, 1545, 1487, 1429, 1368, 1322, 1270, 1237, 1203,
X1186, 1153, 1120, 1106, 1092, 1079, 1065, 1062, 0, 0, 0, 0
X1941, 1810, 1679, 1621, 1562, 1502, 1442, 1382, 1337, 1282, 1246, 1210,
X1192, 1157, 1122, 1106, 1094, 1080, 1066, 1053, 0, 0, 0, 0
X1956, 1825, 1697, 1632, 1572, 1511, 1450, 1396, 1342, 1288, 1251, 1213,
X1195, 1160, 1124, 1118, 1096, 1081, 1065, 1053, 1046, 1038, 1025, 1015,
X1976, 1841, 1796, 1644, 1582, 1520, 1458, 1397, 1350, 1293, 1255, 1217,
| X 910 , 895 , 882 , 864 , 854 , 846 , 828 , 813 , 101 |
| X 1377 , 1247 , 1119 , 1052 , 1008 , 987 , 945 , 916 , 891 |
| X 869 , 857 , 841 , 829 , 818 , 808 , 796 , 781 |
| DATA AR2C/ |
| X 1175 , 1063 , 952 , 900 , 857 , 837 , 798 , 773 , 750 |
| X 727 , 715 , 705 , 696 , 686 , 677 , 667 , 656 , 106 |
| X 1068 , 984 , 880 , 825 , 787 , 770 , 731 , 701 , 678 |
| X 661 , 652 , 638 , 624 , 620 , 610 , 602 , 593 , 108 |
| X 1037 , 928 , 825 , 770 , 737 , 718 , 677 , 650 , 625 |
| X 611 , 600 , 589 , 579 , 575 , 564 , 558 , 551 , 110 |
| X 996 , 899 , 784 , 731 , 692 , 670 , 635 , 607 , 584 |
| X 571 , 562 , 551 , 541 , 535 , 530 , 525 , 519 , 112 |
| X 974 , 874 , 753 , 700 , 677 , 637 , 603 , 571 , 549 |
| X 537 , 526 , 518 , 511 , 505 , 500 , 499 , 494 , 114 |
| X 955 , 853 , 728 , 674 , 632 , 614 , 575 , 542 , 527 |
| X 515 , 504 , 498 , 488 , 483 , 479 , 473 , 470 , 116 |
| X 938 , 834 , 708 , 657 , 613 , 588 , 555 , 529 , 515 |
| X 498 , 486 , 481 , 477 , 474 , 469 , 462 , 457 , 118 |
| X 925 , 819 , 689 , 639 , 598 , 572 , 540 , 517 , 500 |
| X 489 , 479 , 470 , 463 , 468 , 456 , 450 , 448 , 120 |
| DATA AR2D/ |
| X 914 , 810 , 681 , 630 , 588 , 561 , 527 , 507 , 493 |
| X 480 , 473 , 459 , 452 , 458 , 445 , 443 , 438 , 122 |
| X 905 , 798 , 672 , 620 , 580 , 553 , 516 , 498 , 484 |
| X 472 , 464 , 452 , 443 , 442 , 436 , 434 , 430 , 124 |
| X 884 , 789 , 664 , 606 , 572 , 546 , 510 , 491 , 475 |
| X 465 , 456 , 445 , 436 , 435 , 426 , 425 , 421 , 126 |
| X 874 , 781 , 659 , 599 , 564 , 540 , 503 , 484 , 467 |
| X 457 , 448 , 437 , 429 , 426 , 419 , 416 , 411 , 128 |
| X 857 , 770 , 652 , 590 , 550 , 529 , 494 , 472 , 454 |
| X 445 , 435 , 428 , 421 , 414 , 407 , 402 , 395 , 130 |
| X 849 , 762 , 649 , 584 , 543 , 518 , 486 , 461 , 444 |
| X 435 , 424 , 419 , 410 , 405 , 399 , 391 , 384 , 372 |
| DATA AR3/ |
| X 584 , 575 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 |
| X 554 , 537 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 |
| X 513 , 507 , 501 , 500 , 499 , 0 , 0 , 0 , 0 , 0 , 0 , 0 |
| X 490 , 486 , 481 , 480 , 480 , 0 , 0 , 0 , 0 , 0 , 0 , 0 |
| X 451 , 449 , 446 , 449 , 450 , 451 , 454 , 456 , 459 , 463 , 468 , 483 , 487 , 495 , 501 , 147 |
| X 444 , 443 , 442 , 441 , 442 , 443 , 444 , 447 , 450 , 451 , 462 , 472 , 479 , 485 , 491 , 149 |
| X 420 , 416 , 420 , 423 , 425 , 426 , 428 , 431 , 432 , 433 , 444 , 450 , 459 , 468 , 475 , 480 |
| X 410 , 408 , 416 , 420 , 423 , 424 , 425 , 426 , 427 , 431 , 442 , 448 , 455 , 463 , 468 , 475 |

109
NMA=4
IF(JJ.LE.24) GO TO 120
IF(JJ.LT.39) GO TO 140
NMA=3
GO TO 140
120 IF(JJ.GT.22) GO TO 130
NMA=1
GO TO 140
130 IF(JJ.LE.8) GO TO 140
NMA=2
GO TO (150,160,170,220), NMA
150 DGV=AR3(I1,J1)/1000.
GO TO 190
160 I1=I1-7
J1=J1-22
DGV=AR2(I1,J1)/1000.
GO TO 190
170 I1=I1-24
J1=J1-38
DGV=AR3(I1,J1)/1000.
190 CONTINUE
DRAG(M,N)=DGV
200 CONTINUE

C
TR1=RFNL(I)*10.
IF(TR1.LT.0.) TR1=75.
TR2=RENL(I+1)*10.
IF(TR2.LT.0.) TR2=75.
R1=(TR1-RE)/(TR1-TR2)
TR1=AMACH(J)/100.
TR2=AMACH(J+1)/100.
R2=(TR1-AMC)/(TR1-TR2)

210 IF(DRAG(2,1).EQ.0.0.OR.DRAG(1,1).EQ.0.0.OR.DRAG(2,2).EQ.0.0.
X .OR.DRAG(1,2).EQ.0.0) GOTO 220
CD1=R1*(DRAG(2,1)-DRAG(1,1))+DRAG(1,1)
CD2=R1*(DRAG(2,2)-DRAG(1,2))+DRAG(1,2)
CD=R2*(CD2-CD1)+CD1
IFLAG=-1
220 RETURN
220 IRE=RE
WRITE(4,500) AMC,IRE
IFLAG=0
RETURN
500 FORMAT(/38X,20HDRAG VALUE NOT GIVEN,14X,F5.2,I6)
$ASSM
LIST
$FORT
END
$ASSM

DRVT
PROG FINDS DRAG VALUE 14 NOV 79 R01 S REM: DRVT.FOR
CROSS
NORM3
NLSTC
NILST

$FORT
SUBROUTINE DRVT(K, TEMP, RE, AMC, CD, DENTT, HI, LLL)
COMMON /CONST/ RG, PHI, HML, AMS, ZH, DI, VB, AM, GS, AQ, AB
COMMON /CUR/ OMEGA, CP, CPH, CPHZB, CPHSZB, CZ, SPH, SZB
COMMON /THEO/ Z(3), RZ(3), DZ, X(3), RX(3), DX, V(3), RY(3), DY, WX, WY
COMMON /TI/ IMX, BIASHT(600), BIASNY(600), BIASR(600), BIASL(600), JMX, BIASMC(600)
COMMON /ER/ K, I, L, IFLAG
COMMON /DENT/ HI, DENT(I), IFLAG
COMMON /DENT/ HI, DENT(I), IFLAG

HI = RZ(1) + (RX(1) + RY(1) + RX(1) + RY(1)) + RZ(1) / (2. * RG) + HML
Lastly, the routine computes:

**TO BE CONTINUED**

$ASSM

LIST

$FOR1
END

112
$R5-PH

SCRATS,  R 1 3 F T N F I

TO1N FROG CALL S14OOTHFD VEt 1.4

NOV 79

ROtsRMI

KTN

$FOR

SUBROUTINE FITON

COMMON /COEF/ PXY1(51), PXY2(35), PZ2(21), PZ1(35)

COMMON TIME(100), X(100), Y(100), Z(100), TMID(50), XMID(50), YMID(50),

IZ MID(50), XVM(50), YVM(50), ZVM(50), IX, J, N, MIDMAX, KMAX1, KMAX2.

INTEGER*2 IX, J, N, MIDMAX, KMAX1, KMAX2, NIST, NXY1, NXY2

INTEGER*2 I, K, NEND

PX=0.0

PY=0.0

PZX=0.0

PZ=0.0

DO 1 I = 1, KMAX1

PZX=PZX+Z(I)*PXY1(I)

PX=PX+X(I)*PXY1(I)

1 PY=PY+Y(I)*PXY1(I)

NEND=N21+NIST-1

DO 2 I = NIST, NEND

K=I-NIST+1

2 PZ=PZ+Z(I)*PZ1(K)

GO TO 5

ENTRY FITON2

PX=0.0

PZX=0.0

PY=0.0

PZ=0.0

DO 3 I = 1, KMAX1

3 PZ=PZ+Z(I)*PZ1(I)

NEND=NXY1+NIST-1

DO 4 I = NIST, NEND

K=I-NIST+1

4 PY=PY+Y(I)*PXY1(K)

5 TMID(N)=TIME(MIDMAX)

XMD(N)=X(MIDMAX)

YMD(N)=Y(MIDMAX)

ZMD(N)=Z(MIDMAX)

XVM(N)=PX*2.

YVM(N)=PY*2.

ZVM(N)=PZ*2.

ZXM(N)=PZX*2.

RETURN

$ASSM

LIST

$FOR

END
SUBROUTINE INTERP(INTERP, Z2, Z3, IY, IPGE, STALT)

C                             CT/TH, HI, AMC, WE, WN, WT, THETA, PI, RHOOO, T, CD, RE, VRHO, VWX, VYW
C                             VT, INTALT, ITIM, CFREQ, TND1, TND2
C                             TH, ITIM, AND INTALT AS INTEGER+4 VAR

C                             ING, IF=+2 INTERP, IY, IPGE, INCRE

C                             INCRE 200

C                             IF=INTERP EO 2) INCREM=1000

C                             IF=TH greater STALT) GOTO 404

C                             IF INTALT) 700, 700, 701

C                             INTALT IH. INCREM

C                             INCREM INTALT+INCREM

C                             GO TO 40;

C                             IF=INTALT IH) 404, 702, 702

C                             INTALT INTALT

C                             DEIREA (ANTALT-SA1)/(-HI-SA1)

C                             WEIT SA3=DEIREA*(WE-SA3)

C                             WEIT SA4=DEIREA*(WN-SA4)

C                             WEIT SA5=DEIREA*(WT-SA5)

C                             CPU, HANGI (TH, WEIT, WNIT)

C                             IT=INTALT ITI

C                             Z1-I. SA7+DEIREA*(Z2-SA7)

C                             Z1-I. SA8=DEIREA*(Z3-SA8)

C                             CFREQ-0+DEIREA*(CFREQ-SA20)

C                             TN=SA1+DEIREA*(TND1-SA1)

C                             TN=SA2+DEIREA*(TND2-SA2)

C                             VRE=SH=SA17+DEIREA*(VRHO-SA17)

C                             VWX=SA18+DEIREA*(VWX-SA18)

C                             VWY=SA19+DEIREA*(VWW-SA19)

C                             VIT=SA20+DEIREA*(VT-SA20)

C                             VIT=SA29+DEIREA*(VP-SA29)

C                             PINK=SA1=0.01*PI/SA11*DEIREA)

C                             PI=0.0001*SA15+3*(RHOOO/SA13)*DEIREA)

C                             ITS=SA12+DEIREA*(T-SA12)

C                             CDIT=SA14+DEIREA*(CD-SA14)

C                             AMCIT=SA16+DEIREA*(AMC-SA16)

C                             ALT=INTALT

C                             TEMP=ITI

C                             IF(IPGE LT 56) GO TO 33

C                             IPGE=3

C                             WRITE(IY, 401)

C                             IF(INTERP) GO TO 51

C                             FORMAT(IH, 90X, 27H**R M.S. NOISE ERROR IN**3X, 119HTIME ALT .
* NWIND SPEED DIR PRESS T DENSITY Z VEL Z ACC TDE
* CF DENS PRES TEMP EW NW/3X,120HZULU METERS M/S
* M/S M/S DEG MB K GR/M3 M/S M/S
* /S ******PERCENT****** M/S M/S )
33 IPE=PGE+1
WRITE (4,900) AI, T, TEMP, RHOIT, PIMBIT, WEIT, WNIT
900 FORMAT(7,6E10.4)
WRITE (4,327) INTALT, WEIT, WNIT, WIT, IMHIT, PIMBIT, ITIT, RHOIT, Z2IT,
*Z3IT, TND2IT, CFRQIT, VRHOIT, VBIT, ITIT, VXWIT, VWYIT
*3F6.1, 2F5.1)
INTALT=INTALT+INCREM4
GO TO 701
404 SAT=HI
SAT=WE
SAT=WN
SAT=WT
SAT=THETA
SAT=Z2
SAT=Z3
SAT=VP
SAT=VT
SAT=FIMB
SAT=T
SAT=RHO00
SAT=CD
SAT=RE
SAT=AMC
SAT=VRHO
SAT=VWX
SAT=VWy
SAT=CFREQ
SAT=TND1
SAT=TND2
RETURN
$ASSM
LIST
$FORT
END
*ASMM

SCCRT
LINEAR PROG COMPUTE LINEAR COEFFS 30 NOV 79 R01 S REM3:LINEAR FOR
\[ \text{CROSS} \]
\[ \text{NORX3} \]
\[ \text{NLS1C} \]
\[ \text{NLIST} \]
\[ \text{#FORT} \]

SUBROUTINE LINEAR(NPTS)
COMMON /COEF/PXY1(51),PXY2(35),PZ2(21),PZ1(51)
INTEGER NPTS, M, K1, K2, I, M1
TP=NPTS
D=(<TP-1.>*(TP)*(TP+1.))/12.
M=(TP-1.)/2.
M1=M+1
PZ1(M1)=0.
DO 20 I=1,M
 \[ \text{K1=M1+I} \]
 \[ \text{K2=M1+I} \]
PZ1(K2)=I/D
20 PZ1(K1)--PZ1(K2)
RETURN
\[ \text{#ASSM} \]
\[ \text{LIST} \]
\[ \text{#FORT} \]
\[ \text{END} \]
$ASSM

REAVG    PROG  HALF SEC AVERAGING  14 NOV 149 R01   S REM3: REAVG FOR

CROSS
NORX3
NLSTC
NLIST

$FORT

SUBROUTINE REAVG(ICNT, IFLAG3)

COMMON TIME(100), X(100), Y(100), Z(100), TMID(50), XMID(50), YMID(50),
ZMID(50), XMID(50), YMID(50), ZMID(50), IX, J, N, MIDMAX, KMAX1, KMAX2,
XL, NLST1, NXY1, NZ1, ZXM(50)

INTEGER*2 IX, J, N, MIDMAX, KMAX1, KMAX2, NLST1, NXY1, NZ1

INTEGER*2 I, IFLAG3, ICNT

IFLAG3=-1
ST=0.
SX=0.
SY=0.
SZ=0.

DO 1 I=1,5

DATA FROM THE BENCHMARK RUN TAPE Y388
AND PROGRAM B150SERNA*BNCHMRK.PMR
PRODUCES DATA LABELED "TIME X Y Z"

THE FINAL EDITED DATA CORRESPONDING TO
TO "X" (S1) IS USED FOR CALCULATING
Y(J), AND DATA CORRESPONDING TO "Y"
(S2) FOR X(J).

THIS PROGRAM WILL FOLLOW THIS CONVENTION
IN THE STATEMENTS BETWEEN LABELS 5 AND 1.

DATA READ S1=X, S2=Y, S3=Z
READ(1,5,END=2) T, S1, S2, S3

FORMAT(2E18.7)

ST=ST+T
SX=SX+S2
SY=SY+S1
SZ=SZ+S3

TIME(J)=0.2*ST
X(J)=.06096*SX
Y(J)=.06096*SY
Z(J)=.06096*SZ

RETURN
2 IFLAG3=0
RETURN

$ASSM

LIST

$FORT

END
SUBROUTINE RFAVGT(IFLAG4)
COMMON /THFO/2Z(3),RZ(3),DZ,XT(3),RX(3),DX, YT(3),RY(3),
X DY. WX, WY
COMMON TIME(100),X<100>,Y<100>,Z<100>,TMID(50),XMID(50),YMID(50),
A ZMID(50),XVM(50),YVM(50),ZVM(50),IX, J, N, MIDMAX, KMAX1, KMAX2,
B C1, N1ST, NXY1, NZ1, ZXM(50)
INTEGER+2 IX, J, N, MIDMAX, KMAX1, KMAX2, N1ST, NXY1, NZ1
INTEGER+2 I, IFLAG4
IFLAG4:=1
ST:=0.0
SX:=0.0
SY:=0.0
SZ:=0.0
DO I=1,5
CALL THFOT(T)
IF(T. EQ. -999.9) GO TO 10
ST:=ST*T
SX:=SX+XT(I)
SY:=SY+YT(I)
1 ST:=ST+S2+ZT(I)
TIME(J)=0.2*ST
X(J)=0.2*SX
Y(J)=0.2*SY
Z(J)=0.2*S2
RETURN
10 IFLAG4:=0
RETURN
ASSM LIST hi YRT
*ASSM

SIDE  SCRT
PROG  MOVE DATA 1 PT  14 NOV 79 R01 5 RFM3: SLIDF.FOR
CROSS
NORX3
NLSTC
NLST

*FOR

SUBROUTINE SLIDE(XR, ICNT, IFLAG)
COMMON TIME(100), X(100), Y(100), Z(100), TH(50), XM(50), YM(50),
A  ZM(50), XVM(50), YVM(50), ZVM(50), IX, J, N, MIDMAX, KMAX1, KMAX2.
B  C, NIST, NXY1, NZ1, ZXM(50)
C  INTEGRATE X, J, N, MIDMAX, KMAX1, KMAX2, NIST, NXY1, NZ1
D  INTEGRATE I, IA, IC, ICNT, IFLAG3, IFLAG5
E  IFLAG5=-1
F  IA=KMAX1-IP
G  DO 110 I=1, IA
H  IC=I+IB
I  X(I)=X(IC)
J  Y(I)=Y(IC)
K  Z(I)=Z(IC)
L  110 TIME(I)=TIME(IC)
M  IC=IC+1
N  DO 111 J=IC, KMAX1
O  CALL REAVG(ICNT, IFLAG3)
P  IF(IFLAG3)114, 112, 113
Q  114 RETURN
R  112 IFLAG=0
S  RETURN
T  113 IFLAG=1
U  RETURN

*ASSM

LIST

*FOR

END
$ASMN
SCRAT
SLIDE
MOVE TDATA 1 PT 14 NOV 79 R01. S REM3:SLIDET.FOR
CROSS
NOSC
LIST
LIST
*FORT
SUBROUTINE SLIDET(IB, IFLAG6)
COMMON TIME(100), X(100), Y(100), Z(100), TMD(50), XMID(50), YMID(50),
A ZMID(50), XMID(50), YMID(50), ZMID(50), IX, J, N, MIDMAX, KMAX1, KMAX2,
B C1, N1ST, NXY1, NXY1, ZMX(50)
INTEGER+2 IX, J, N, MIDMAX, KMAX1, KMAX2, N1ST, NXY1, NZ1
INTEGER+2 IA, IC, IFLAG4, IFLAG6
IFLAG6= -1
IA= KMAX1-IB
DO 10 I= IA, IA
10 IC= IA+IB
X(I)= X(IC)
Y(I)= Y(IC)
Z(I)= Z(IC)
IFLAG4= GT(IA+1)
DO 111 J= IA, JMAX1
111 CALL RPA(XT(YT(ZT))(IFLAG4), IFLAG4)
IF(IFLAG4)+1, 112, 113
113 RETURN
112 IFLAG6= 0
RETURN
*ASMN
LIST
*FORT
END
$ASSM

THB PRG CONSTR PRNT PLOT 14 NOV 79 R01 $ RM3: TAB FOR

CROSS
NORX3
NLIC
NLIST

$FORT

SUBROUTINE TAR(SALT, IV)

INTEGER*2 I, IV, J
DIMENSION DEN(5,8), WIND(5,8)
DATA ZERO/* -- */

DATA DEN / 3*4H0. 01, 4H0. 47, 4H0. 91,
2*4H0. 01, 4H0. 82, 4H0. 51, 4H0. 95 ,
2*4H0. 01, 4H0. 03, 4H0. 59, 4H0. 96 ,
4H0. 01, 4H0. 04, 4H0. 47, 4H0. 72, 4H0. 98 ,
4H0. 01, 4H0. 22, 4H0. 84, 4H0. 97, 4H0. 99 ,
4H0. 22, 4H0. 52, 4H0. 90, 4H0. 98, 4H0. 99 ,
4H0. 42, 4H0. 88, 4H0. 96, 4H0. 99, 4H0. 99 ,
4H0. 75, 4H0. 96, 4H0. 99, 4H0. 99, 4H0. 99 ,
DATA WIND / 4H0. 01, 4H0. 02, 4H0. 03, 4H0. 14 , 4H0. 58,
4H0. 01, 4H0. 03, 4H0. 04, 4H0. 17, 4H0. 63 ,
4H0. 01, 4H0. 04, 4H0. 06, 4H0. 24, 4H0. 66 ,
4H0. 01, 4H0. 06, 4H0. 18, 4H0. 85, 4H0. 90 ,
4H0. 01, 4H0. 10, 4H0. 76, 4H0. 97, 4H0. 99 ,
4H0. 02, 4H0. 40, 4H0. 98, 4H0. 99, 4H0. 99 ,
4H0. 57, 4H0. 76, 4H0. 99, 4H0. 99, 4H0. 99 ,
4H0. 96, 4H0. 99, 4H0. 99, 4H0. 99, 4H0. 99 ,
100 FORMAT(1H1, 29X, 68XRATIO OF AMPLITUDE OF SMOOTHED DENSITY WAVE TO A
AMPLITUDE OF ORIGINAL/32X, 46XWAVE AS A FUNCTION OF ALTITUDE AND WAV
2ELENGTH/29X, 50XALITUDE AND WAVELENGTH MEASURED IN KILOMETERS. )
101 FORMAT(20X, 62XRATIO OF AMPLITUDE OF SMOOTHED SINUSOIDAL WIND TO AN
AMPLITUDE OF/23X, 55XORIGINAL WIND AS A FUNCTION OF ALTITUDE AND WAV
2ELength. /29X, 50XALITUDE AND WAVELENGTH MEASURED IN KILOMETERS. )
102 FORMAT(46X, 8HALITUDE//26X, 57HX 100 X 90 X 80 X 70 X 60 X
1 50 X 40 X 30 X/26X, 57X(1HX)/20X, 7HE X. 8(6X, 1HX)/20X, 7HA ,
2 1 X, 8(A6, 1HX)/20X, 7HV X. 8(6X, 1HX)/20X, 7HE 2 X, 8(A6, 1HX)/20
3X, 7HL X. 8(6X, 1HX)/20X, 7HE 5 X, 8(A6, 1HX)/20X, 7HN X. 8(6X, 1
4X)/20X, 7HG 10 X, 8(A6, 1HX)/20X, 7HT X. 8(6X, 1HX)/20X, 7HE 20 X,
58(A6, 1HX)/////////)
WRITE(IV, 100)
IF(SALT, GT, 75.) GO TO 20
10 WRITE(IV, 10?) (ZERO, ZERO, ZERO, ZERO, (DEN(I, J), J=5, 8), I=1, 5)
WRITE(IV, 101)
WRITE(IV, 102) (ZERO, ZERO, ZERO, ZERO, (WIND(I, J), J=5, 8), I=1, 5)
RETURN
20 IF(SALT, GT, 90.) GO TO 30
WRITE(IV, 102) (ZERO, ZERO, ZERO, (DEN(I, J), J=4, 8), I=1, 5)
121
51 WRITE(IY,101)
52 WRITE(IY,102) (ZERO, ZERO, ZERO, WIND(I, J), J=4,8), I=1,5)
53 RETURN
54 30 IF(ESLT.GT.100.) GO TO 40
55 WRITE(IY,102) (ZERO, ZERO, DFN(I, J), J=3,8), I=1,5)
56 WRITE(IY,101)
57 WRITE(IY,102) (ZERO, ZERO, WIND(I, J), J=3,8), I=1,5)
58 RETURN
59 40 IF(ESLT.GT.110.) GO TO 50
60 WRITE(IY,102) (ZERO, DFN(I, J), J=2,8), I=1,5)
61 WRITE(IY,101)
62 WRITE(IY,102) (ZERO, WIND(I, J), J=2,8), I=1,5)
63 RETURN
64 50 WRITF(IY,102) (DFN(I, J), J=1,8), I=1,5)
65 WRITE(IY,101)
66 WRITE(IY,102) (WIND(I, J), J=1,8), I=1,5)
67 RETURN
68 $ASSM
69 LIST
70 $FORT
71 END
SUBROUTINE THFOT(T)

INTEGER*2 I1, K, U, IU
COMMON /CONST/ RG, PHI, HMSG, AMS, ZB, DI, VB, AM, GSRG, AB
COMMON /THEO/ Z(3), RZ(3), DZ, X(3), RX(3), DX, Y(3), RY(3), DY, WX, WY
DIMENSION A(4), B(4), PZ(2), PX(2), PY(2), QZ(2), QX(2), QY(2)
DATA A..0. 0, 0. 5, 0. 5, 1. 0, 1. 0, 2. 0, 2. 0, 1. 0, IU/0/
IF(IU. NE..0) GO TO 2
T=0.
IU=1.
2 DO 1 K=1, 2
PZ(K)=0. 0
PX(K)=0. 0
PY(K)=0. 0
QZ(K)=0. 0
QX(K)=0. 0
QY(K)=0. 0
1 QV(K)=0. 0
H=0. 1
DO 6 U=1, 4
DO 7 K=1, 2
RZ(K)=Z(K)+A(U)*PZ(K)
RX(K)=X(K)+A(U)*PX(K)
7 RY(K)=Y(K)+A(U)*PY(K)
DO 6 K=1, 2
CALL DRVTCKJ At., RP, A3. A4, A5, 86, 11)
PZ(K)=DZ*H
PX(K)=DX*H
PY(K)=DY*H
QZ(K)=QZ(K)+B(U)*PZ(K)
QX(K)=QX(K)+B(U)*PX(K)
QY(K)=QY(K)+B(U)*PY(K)
6 CONTINUE
DO 8 K=1, 2
Z(K)=Z(K)+QZ(K)/6.
X(K)=X(K)+QX(K)/6.
8 Y(K)=Y(K)+QY(K)/6.
44 T=T+H
TALT=Z(1)+X(1)*X(1)+Y(1)*Y(1)/(2*RG)+HMSG
IF(TALT.GT.55000.) GO TO 10
47 T=-999. 9
IU=0
10 RETURN
$ASSM
LIST
$ASSM
END
SUBROUTINE TIFALL(HI, TFT, IFT, ZVM, HI2, COLAPS, IPGE, IY)
INTEGER*2 IFT, HI2, IPGE, IY, JFT
COMMON /TF/, TFI, TFT2, TFT3, TFT4, TFT5, TFT6, TFT7, TFT8, TFT9, TFT10
DIMENSION ZVM(50)
302 IF(ZVM(2).LT.ZVM(1)) GO TO 303
312 IFT=5
313 JFT=1
314 WRITE(IY, 411)
315 FORMAT(20X, 50HAPOGEE NOT KNOWN. TIME OF FALL TEST NOT USED ABOVE 5
316 X KM. )
317 IFGE=IPGE+1
318 GO TO 300
320 IF (HI-77900. ) 350, 304, 304
323 304 IF (ZVM(1)+200. ) 412, 305, 305
324 305 IFT=3
325 GO TO 399
326 IF (ZVM(1)-ZVM(2)) 412, 412, 307
327 307 IF (ZVM(1)+200. ) 308, 308, 399
328 308 IFT=5
329 HI2=HI/1000. +2. 1
330 IH2=HI2
331 HI3=HI2
332 IF (HI2-HI3-. 5) 309, 310, 310
333 309 HI2=IH2
334 GO TO 311
335 310 HI2=IH2+1
336 311 H2=HI2
337 WRITE(IY, 411) HI2
338 413 FORMAT(1H, 20X, 16H5BALLOON APOGEE =, I4, 4H KM. )
339 IPGE=IPGE+1
340 IFT=6
341 RETURN
342 312 IF(TFT2. EQ. 0. 0) RETURN
343 IF (HI2-100. ) 300, 301, 313
344 313 IF(HI2-125. ) 314, 318, 319
345 314 IF(HI2-115. ) 315, 316, 317
346 310 IFT=7
347 GO TO 325
348 301 TTEST=29.
349 GO TO 323
350 315 TTEST=25. +4. *(115. -HI2)/15.
GO TO 323  
316 TTEST=25.  
GO TO 323  
GO TO 323  
318 TTEST=24.  
GO TO 323  
319 IF(HI2=140. ) 320, 321, 322  
320 TTEST=22. +(40. -H12)/15.  
GO TO 323  
321 TTEST=22.  
GO TO 323  
323 IF(TFT. LT. TTEST+3. . AND. TFT. GT. TTEST-3. ) GO TO 324  
WRITE(IY, 414)  
414 FORMAT(20X,76HTIMF OF FAIL TEST BETWEEN 80 AND 70 KM. INDICATES BA  
*LOON DID NOT INFLATE. )  
IF(COLAPS. EQ. 0. ) COLAPS=70000.  
IPGE=IPGF+1  
IFT=7  
RETURN  
324 WRITE(IY, 415)  
415 FOKMAT(20X, 32HBALLOON STILL INFLATED AT 70 KM. )  
IPGE=IPGE+1  
IFT=7  
RETURN  
325 IF(TFT3. EQ. 0. 0) RETURN  
IF(TFT. LT. 52. . AND. TFT. GT. 44. ) GO TO 326  
WRITE(IY, 416)  
416 FORMAT(20X, 57HTIMF OF FAIL TEST BETWEEN 70 AND 60 KM. INDICATES CO  
AL APSE)  
IF(COLAPS. EQ. 0. ) COLAPS=60000.  
IPGE=IPGF+1  
IFT=8  
RETURN  
350 IFT=4  
351 IF(ZVM(1). LT. -150. ) GO TO 355  
IFT=11  
RETURN  
351 IF(ZVM(2). GT. ZVM(1)) GO TO 412  
IF(ZVM(1). GT. -150. ) RETURN  
IFT=8  
H12=HI/1000. +1. 5  
IM2=H12  
IM3=H12
IF(HI2-HI3-.5) 352, 353, 353
352 HI2=IH2
GO TO 354
353 HI2=IH2+1
IH2=HI2
WRITE(IY, 413) IH2
IPGE=IPGE+1
GO TO 360
355 IF(ZVM(1).LT.-210.) GO TO 412
360 IF(TFT4.EQ.0.0) RETURN
361 IF(TFT.LT.39. .AND. TFT.GT.31.) GO TO 361
362 IF(TFT5.EQ.0.0) RETURN
363 WRITE(IY, 419)
364 GO TO (362, 365, 367, 369, 371), JFT
365 IF(TFT5.EQ.0.0) RETURN
366 GO TO 366
367 IF(TFT4.EQ.0.0) RETURN
368 WRITE(IY, 419)
369 IF(TFT4.EQ.0.0) RETURN
370 WRITE(IY, 419)
371 GO TO 371
372 IF(TFT5.EQ.0.0) RETURN
373 WRITE(IY, 419)
399 RETURN
365 IF (TFT. LE. 0. 0) GO TO 399
JFT=3
IPGE=IPGF+1
IF(TFT.LT.74. AND. TFT.GT.60. ) GO TO 366
WRITE(IY,422)
422 FORMAT(20X,57HTIME OF FALL. TEST BETWEEN 50 AND 45 KM. INDICATES CO
*LAPSE )
IF(COLAPS. EQ. 0. 0) COLAPS=45000.
GO TO 399
366 WRITE(IY,423)
423 FORMAT(20X,32HBALLOON STILL INFLATED AT 45 KM. )
GO TO 399
367 IF (TFT8. LE. 0. 0) GO TO 399
JFT=4
IPGE=IPGF+1
IF(TFT.LT.106. AND. TFT.GT.84. ) GO TO 368
WRITE(IY,424)
424 FORMAT(20X,57HTIME OF FALL. TEST BETWEEN 45 AND 40 KM. INDICATES CO
*LAPSE )
IF(COLAPS. EQ. 0. 0) COLAPS=40000.
GO TO 399
368 WRITE(IY,425)
425 FORMAT(20X,32HBALLOON STILL INFLATED AT 40 KM. )
GO TO 399
369 IF (TFT9. LE. 0. 0) GO TO 399
JFT=5
IPGE=IPGE+1
IF(TFT.LT.124. AND. TFT.GT.124. ) GO TO 370
WRITE(IY,426)
426 FORMAT(20X,57HTIME OF FALL. TEST BETWEEN 40 AND 35 KM. INDICATES CO
*LAPSE )
IF(COLAPS. EQ. 0. 0) COLAPS=35000.
GO TO 399
370 WRITE(IY,427)
427 FORMAT(20X,32HBALLOON STILL INFLATED AT 35 KM. )
GO TO 399
371 IF (TFT0. LE. 0. 0) GO TO 399
JFT=1
IPGE=IPGE+1
IF(TFT.LT.232. AND. TFT.GT.198. ) GO TO 372
WRITE(IY,428)
428 FORMAT(20X,57HTIME OF FALL. TEST BETWEEN 35 AND 30 KM. INDICATES CO
*LAPSE )
IF(COLAPS. EQ. 0. 0) COLAPS=30000.
GO TO 399
372 WRITE(IY,429)
429 FORMAT(20X,32HBALLOON STILL INFLATED AT 30 KM. )
GO TO 399
200 $ASSM
201 LIST
202 $FORT
203 END
127
$ASSM

SCRAT

TIFAL 2  FROG CHK SPH2 COLLAPSE 14 NOV 79 R01  S REM3:TIFAL2.FOR

CROSS

NORX3

NLSTC

NLIST

$FORT

SUBROUTINE TIFAL 2(HI, TFT, IFT, ZVM, HI2, COLAPS, IPGE, IV)

INTEGER IFT, JFT, HI2, IPGE, IV

COMMON ,TF/TFT1, THT2, TFT3, TFT4, TFT5, THT6, TFT7, TFT8, TFT9, TFT10

DIMENSION ZVM(58)

GO TO (399, 302, 306, 350, 464, 312, 325, 360, 356, 362, 351), IFT

302 IF(ZVM(2).LT.ZVM(1)) GO TO 303

412 IFT=5

JFT=1

WRITE(IV,411)

411 FORMAT(20X,55S8APOGEE NOT KNOWN. TIME OF FALL TEST NOT USED ABOVE 5 +5 KM )

IPGE=IPGE+1

GO TO 399

303 IF (HI-77300 ) 350, 304, 304

304 IF (ZVM(1)+200. ) 412, 305, 305

305 IFT=3

GO TO 399

306 IF (ZVM(1)-ZVM(2)) 412, 412, 307

307 IF (ZVM(1)+200.) 308, 308, 399

308 IFT=5

HI2=HI/1000. +? 1

IHC=HI?

HI3=IH2

IF (HI3-HI3-5) 309, 310, 310

309 HI2=HI2

GO TO 311

310 HI2=IH2+1.

IH2=HI2

311 WRITE(IV,413) IH2

413 FORMAT(1H ,20X, 16HBALLOON APOGEE =, 14, 4H KM. )

IPGE=IPGE+1

IFT=6

RETURN

312 IF(TFT2.EQ.0.0) RETURN

TTEST=(-23.

IF(TFT.LT. TTEST+5. .AND. TFT.GT. TTEST-5.) GO TO 324

WRITE(IV,414)

414 FORMAT(20X,76HTIME OF FALL TEST BETWEEN 80 AND 70 KM. INDICATES BA

*LLOON DID NOT INFLATE. )

IF(COLAPS. EQ. 0.) COLAPS=70000.

IPGE=IPGE+1

IFT=?
51 RETURN
52 324 WRITE(IY, 415)
53 415 FORMAT(20X, 32HBALLOON STILL INFLATED AT 70 KM. )
54 IPGE=IPGE+1
55 IFT=7
56 RETURN
57 325 IF(TFT3.EQ. 0.) RETURN
58 IF(TFT.LT. 46. AND. TFT. GT. 38.) GO TO 326
59 WRITE(IY, 416)
60 416 FORMAT(20X, 37HTIME OF FALL TEST BETWEEN 70 AND 60 KM. INDICATES CO
61 ALAPSE)
62 IF(COLAPS. EQ. 0. ) COLAPS=60000.
63 IFG=IPGE+1
64 IFT=8
65 RETURN
66 326 WRITE(IY, 417)
67 417 FORMAT(20X, 32HBALLOON STILL INFLATED AT 60 KM. )
68 IPGE=IPGE+1
69 IFT=8
70 RETURN
71 350 IFT=4
72 IF(ZVM(1). LT. -150. ) GO TO 355
73 IF=11
74 RETURN
75 351 IF(ZVM(2). GT. ZVM(1)) GO TO 412
76 IF(ZVM(1). GT. -150. ) RETURN
77 IFT=8
78 HI2=HI/1000. +1. 5
79 IH2=HI2
80 HI3=IH2
81 IF(HI2-HI3-. 5) 352, 353, 353
82 352 HI2=IH2
83 GO TO 354
84 353 HI2=IH2+1
85 IH2=HI2
86 354 WRITE(IY, 413) IH2
87 IPGE=IPGE+1
88 GO TO 360
89 355 IF(ZVM(1). LT. -210. ) GO TO 412
90 IFT=9
91 RETURN
92 356 IF(ZVM(2). GT. ZVM(1)) GO TO 412
93 IF(ZVM(1). GT. -210. ) RETURN
94 HI2=HI/1000. +2. 1
95 IH2=HI2
96 HI3=IH2
97 IF(HI2-HI3-. 5) 357, 358, 358
98 357 HI2=IH2
99 GO TO 359
100 358 HI2=IH2+1

129
101       IH2=HI2
102       359 WRITE(IV,413) IH2
103       IPGF=IPGE+1
104       360 IF(TFT4.EQ.0.0) RETURN
105       IFT=10
106       IF(TFT.LT.35.0 AND. TFT.GT.27.) GO TO 361
107       WRITE(IV,418)
108       418 FORMAT(20X,57HTIME OF FALL TEST BETWEEN 60 AND 55 KM. INDICATES CO
109          ALAPSE )
110       IF(COLAPS.EQ.0.) COLAPS=55000.
111       IPGE=IPGF+1
112       RETURN
113       361 WRITE(IV,419)
114       419 FORMAT(20X,32HBALLOON STILL INFATED AT 55 KM. )
115       IPGE=IPGF+1
116       RETURN
117       464 GO TO (362,365,367,369,371),JFT
118       362 IF(TFT5.EQ.0.0) RETURN
119       IFT=1
120       JFT=2
121       IF(TFT.LT.47.0 AND. TFT.GT.36.) GO TO 363
122       WRITE(IV,420)
123       420 FORMAT(20X,57HTIME OF FALL TEST BETWEEN 55 AND 56 KM. INDICATES CO
124          ALAPSE )
125       IF(COLAPS.EQ.0.) COLAPS=50000.
126       IPGE=IPGF+1
127       RETURN
128       367 WRITE(IV,421)
129       421 FORMAT(20X,32HBALLOON STILL INFATED AT 50 KM. )
130       IPGE=IPGF+1
131       RETURN
132       368 IF(TFT3.EQ.0.0) GO TO 399
133       JFT=3
134       IPGE=IPGF+1
135       IF(TFT.LT.65.0 AND. TFT.GT.50.) GO TO 366
136       WRITE(IV,422)
137       422 FORMAT(20X,57HTIME OF FALL TEST BETWEEN 50 AND 45 KM. INDICATES CO
138          ALAPSE )
139       IF(COLAPS.EQ.0.) COLAPS=45000.
140       GO TO 399
141       366 WRITE(IV,423)
142       423 FORMAT(20X,32HBALLOON STILL INFATED AT 45 KM. )
143       GO TO 399
144       369 IF(TFT4.EQ.0.0) GO TO 399
145       JFT=4
146       IPGE=IPGF+1
147       IF(TFT.LT.93.0 AND. TFT.GT.72.) GO TO 368
148       WRITE(IV,424)
149       424 FORMAT(20X,57HTIME OF FALL TEST BETWEEN 45 AND 40 KM. INDICATES CO
150          ALAPSE )
IF(COLAPS.EQ.0.0) COLAPS=40000.

GO TO 399

WRITE(IY,425)

FORMAT(20X,32X,BALLOON STILL INFLATED AT 40 KM.)

GO TO 399

CONTINUE

CONTINUE

GO TO 399

$ASSM  LIST

$FORT  END
```plaintext
SUBROUTINE TROBINKGS, N2MID, KSW2, KSW, IPGF, IV)
INTERGER+2 ICJ, IFLAG, IFLAG1, IFLAG6, IPGE, ISN, IV
INTERGER+2 KSN, KSN2, NPRT, N2MID, JB, IMX, JMK
INTERGER+2 IX, JN, MIDK4, KM1X, KM2X, N1ST, NX1, NZ1
INTERGER+2 I, KX, NKX2, NZP, NX1MID, NX2MID, NZ1MID, NZ2MID, N2ST
COMMON /FIRST/ NX1, NZ1, NX2, N1ST, NX1MID, NX2MID, NZ1MID, NZ2MID, N2ST
COMMON /CODRES/ PXX1(5), PXX2(5), P22(21), P11(51)
COMMON /VARUST/ RG, ALA, KSFL, AMS, ZR, DIA, VB, AM, GSRG, AR
COMMON /ACOR12/ TOSL, TOCl, TOCL2, TOCL3
COMMON /CONC2/ AMK, ALPHA, FU, RAD
COMMON /TARLU/ IMX, BIASHI(600), BIASNH(600), BIASNY(600),
                 X BIASK(600), BIAST(600), JMK, BIASMC(600)
COMMON /TIME(1000), X(1000), Y(1000), Z(1000), TH1D(50), XM1D(50),
                 X Y1D(50), Z1D(50), XM2D(50), Y1M(50), Z1M(50), IY, JN.
                 X MIDMAX, KM1X, KM2X, CI, N1ST, NX1,
                 X NZ1, Z2M(50)

DO 30 IX = 1, KMAX1
   CALL REMGT(IFLAG1)
   IF(IFLAG1)6, 90, 6
   6 GO TO 7, 9, KSH2
   7 GO TO K-1, KMAX2
   9 N K
   CALL FR1ON
   CALL SLIDE(2, IFLAG6)
   IF(IFLAG6)8, 90, 8
   8 CONTINUE
   GO TO 11
   9 DO 16 K = 1, KMAX2
      11 N K
      CALL FR1ON2
      CALL SLIDE(2, IFLAG6)
      IF(IFLAG6)16, 90, 10
   10 CONTINUE
   11 KSH = 0, 6
   16 ZSH = 0, 6
   19 YS = 0, 6
   22 ZS = 0, 6
```

132
GO TO (12, 15), KSN
12 DO 13 I=1, NXY2
  13 Y3=V3+FXY2(I)+YVM(I)
  NEND=N2ST+N72 -1
  DO 14 I=N2ST, NEND
  K=I-N2ST+1
14 Z3=Z3+P22(K)+ZVM(I)
  GO TO 18
15 DO 16 I=1, NZ2
  16 Z3=Z3+P22(I) +ZVM(I)
  NEND=NXY2-N2ST -1
  DO 17 I=1-N2ST, NEND
  K=I-N2ST+1
  Z3=Z3+FXY2(K)+ZVM(I)
  X3=X3+FXY2(K)+XVM(I)
17 Y3=V3+FXY2(K)+YVM(I)
18 CONTINUE
  X2=XVM(N2MID)
  Y2=YVM(N2MID)
  Z2=ZKM(N2MID)
  X1=XMID(N2MID)
  Y1=YMID(N2MID)
  Z1=ZMID(N2MID)
  T11=TMID(N2MID)
  HI=Z1+(X1*X1+Y1*Y1)/(2.*RG)+HMSL
  OPHOR=1.*HI/SH
  WTHR=GS/(RG+OPHOR+OPHOR+OPHOR)
  IF(Z3.LT.-8.0) GO TO 2050
  CORX=TOCSZ*X2+TOSL*Y2
  CORY=TOCLSZ*X2-TOCSL*Y2
  CORZ=TOCLSZ*X2-TOCLS*Y2
  GO TO (19, 38), ISN
19 CALL ATMOS(HI, T, RHO00)
  ICD=5
  VT=ABS(Z2)*SORT(1.0+(X3*X3+Y3*Y3)/(Z3+WTHR+Z1+RG)**2)
  PE=DIAM*RHO00+VT*(T+T*T)/(0.000001458+SORT(T*T))
  AMC=VT*(SORT(288.16/T))/340.29705
  CALL DRAFT(CD, AMC, RE, IFLAG)
  IF(IFLAG)34, 2000, 34
  34 ICD=1
  CD1=CD
  RHO00=RHO00
  RHO=RHO00
  GO TO 20
  38 DCD=CD
  CD=2.0*CD-CD1
100 CD1=DCD
101  20 WTHO=(1.0-VB+RHO/RMS)*WTHR
102  BOUYX=X1+WTWO
103  BOUYV=Y1+WTWO
104  BOUYZ=(Z1+RG)+WTWO
105  21 WDENOM=2.0*X1+COR2-BOUYZ
106  22 WX=WZ1
107  WX=WZ2-(Z2-WZ2)*(X3+CORX-BOUYX)/WDENOM
108  WY=WZ2-(Z2-WZ2)*(Y3+CORY-BOUYV)/WDENOM
109  WZ1=(X1+WY+Y1+WX)/(HI+RG)
110  IF(ARS(WZ1, WZ1) GT. 0.1) GO TO 22
111  WTH=0.1
112  VT=SORT((Y2-WX)**2+(Y2-WY)**2+(Z2-WZ)**2)
113  FHE. AHS=/=COR2-23+WTHR*(RG+21)/(CD+AR+VT*(Z2-WZ)+WTHR*(RG+21)*VB)
114  GO TO (23, 24), ISN
115  23 PI=1+RHO/RMS
116  GO TO 25
117  24 PI=PI+0.5+RHO*2+(HI-HII)*EXP(0.5+RHO*(RHO+RHO0))+(RHO+0.5*(HI+HII)
118  RHO/RMS)
119  T=PI+AMK/(RHO+RHO)
120  25 AMC=VT*(SORT(T1)**2)/1000.29205
121  IF=1/RHO*VT*(T+11.9)/(0.5+9.00006458+SORT(T1**2))
122  CD=CD
123  CALL DRAGT(CD, AMC, RE, IFLAG)
124  IF(IFLAG=23, 26, 33)
125  31 CD=1
126  IF(NFRT EQ. 6) GO TO 26
127  CD=CD+CD-CD
128  IF(NPRT/RHO-RHO0/RHO, LT. 0.003333) GO TO 28
129  IF(NFRT EQ. 6) GO TO 28
130  26 RHO0=RHO
131  NPRT, NPRT+1
132  GO TO 26
133  27 RHO = (RHO + RHO0)/2.0
134  CD=CD+CD/2.0
135  NPRT=0
136  GO TO 26
137  28 RHO00=RHO
138  NPRT=0
139  RHO0=RHO
140  PI=PI
141  HI=HI
142  IF(Z3, LT. -8.0) GO TO 2650
143  ISNH=2
144  CALL ATMOS(HI, TSS, RSS)
145  JB=JB+1
146  IF(JB, GT. 660) GO TO 2660
147  BIASI(JB)=HI
148  BIASMO(JB)=AMC
149  BIASNX(JB)=WX
150  BIASNY(JB)=WY
BIASR(JB)=RHO-RSS
BIAST(JB)=T-TSS
JMX=JB
GOTO 2000
2000 ICD=ICD+1
IF(ICD.LE.5) GO TO 2050
ICD=1
ISW=1
2050 CONTINUE
NPRT=0
32 K=KMAX2-1
DO 31 I=1,K
XMID(I)=XMID(I+1)
YMID(I)=YMID(I+1)
ZMID(I)=ZMID(I+1)
TMID(I)=TMID(I+1)
XVM(I)=XVM(I+1)
YVM(I)=YVM(I+1)
ZVM(I)=ZVM(I+1)
31 ZVM(I)=ZVM(I+1)
GO TO (29, 30), KSW2
29 N=KMAX2
CALL FITON
CALL SLIDE1(2, IFLAG6)
IF(IFLAG6)11, 96, 11
30 N=KMAX2
CALL FITON2
CALL SLIDE1(2, IFLAG6)
240 IF(IFLAG6)11, 96, 11
96 CONTINUE
RETURN
#ASSM
LIST
#FORT
END
$ASSM
SCRAT
WANGI. PROG POLAR WIND COORD 14 NOV 79 R61 S RFM3:WANG1.FOR
CROSS
NORX3
NLSIC
NLIST

$FORT

SUBROUTINF WANGL(THETA, WF, WN)

INTEGER*2 I, NEND, N2MID
RAD=57.2957795
IF (WF)1003, 1002, 1001
1001 IF(WN)1011, 1011, 1010
1002 IF(WN)1014, 1014, 1010
1003 IF(WN)1013, 1012, 1012
1010 TH=3.1415927+ATAN (WF/WN)
GO TO 1018
1011 TH=4.7123898+ATAN (-WN/WF)
GO TO 1018
1012 TH=1.5707963+ATAN (-WN/WE)
GO TO 1018
1013 TH=ATAN (WF/WN)
GO TO 1018
1014 TH=0.0
1018 THETA= (TH)*(RAD)
RETURN
$ASSM
LIST
$FORT
END

136
VI. REFERENCES
