NOAA Technical Report ERL 281-APCL 29

U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Environmental Research Laboratories

Computer Simulation of Multi-Aircraft Flights: An Aid to Mission Planning

STEPHEN D. WHITAKER

OULDER, COLO.
OCTOBER 1973

ENVIRONMENTAL RESEARCH LABORATORIES

The mission of the Environmental Research Laboratories is to study the oceans, inland waters, the lower and upper atmosphere, the space environment, and the earth, in search of the understanding needed to provide more useful services in improving man's prospects for survival as influenced by the physical environment. Laboratories contributing to these studies are:

Earth Sciences Laboratories (ESL): Seismology, geomagnetism, geodesy, and related earth sciences; earthquake processes, internal structure and shape of the earth and distribution of the earth's mass.

Atlantic Oceanographic and Meteorological Laboratories (AOML): Geology and geophysics of ocean basins, oceanic processes, and sea-air interactions (Miami, Florida).

Pacific Oceanographic Laboratories (POL): Oceanography with emphasis on the oceanic processes and dynamics; tsunami generation, propagation, modification, detection, and monitoring (Seattle, Washington).

Atmospheric Physics and Chemistry Laboratory (APCL): Processes of cloud and precipitation physics; chemical composition and nucleating substances in the lower atmosphere; and laboratory and field experiments toward developing feasible methods of weather modification.

Air Resources Laboratories (ARL): Diffusion, transport, and dissipation of atmospheric contaminants; development of methods for prediction and control of atmospheric pollution; geophysical monitoring for climatic change (Silver Spring, Maryland).

Geophysical Fluid Dynamics Laboratory (GFDL): Dynamics and physics of geophysical fluid systems; development of a theoretical basis, through mathematical modeling and computer simulation, for the behavior and properties of the atmosphere and the oceans (Princeton, New Jersey).

National Severe Storms Laboratory (NSSL): Tornadoes, squall lines, thunderstorms, and other severe local convective phenomena directed toward improved methods of prediction and detection (Norman, Oklahoma).

Space Environment Laboratory (SEL): Solar-terrestrial physics, service and technique development in the areas of environmental monitoring and forecasting.

Aeronomy Laboratory (AL): Theoretical, laboratory, rocket, and satellite studies of the physical and chemical processes controlling the ionosphere and exosphere of the earth and other planets, and of the dynamics of their interactions with high altitude meteorology.

Wave Propagation Laboratory (WPL): Development of new methods for remote sensing of the geophysical environment with special emphasis on optical, microwave and acoustic sensing systems.

Weather Modification Program Office (WMPO): Plans and directs ERL weather modification activities, operates ERL aircraft fleet, and research on cumulus cloud modification, on hurricanes and other tropical problems, and on hurricane modification.

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION

BOULDER, COLORADO 80302

U.S. DEPARTMENT OF COMMERCE
Frederick B. Dent, Secretary

NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
Robert M. White, Administrator
ENVIRONMENTAL RESEARCH LABORATORIES
Wilmot N. Hess, Director

NOAA TECHNICAL REPORT ERL 281-APCL 29

Computer Simulation of Multi-Aircraft Flights: An Aid to Mission Planning

STEPHEN D. WHITAKER

BOULDER, COLO.
October 1973

For sale by the Superintendent of Documents, U. S. Government Printing Office, Washington, D. C. 20402

DISCLAIMER

The Environmental Research Laboratories do not approve, recommend, or endorse any proprietary product or proprietary material mentioned in this publication. No reference shall be made to the Environmental Research Laboratories or to this publication furnished by the Environmental Research Laboratories in any advertising or sales promotion which would indicate or imply that the Environmental Research Laboratories approve, recommend, or endorse any proprietary product or proprietary material mentioned herein, or which has as its purpose an intent to cause directly or indirectly the advertised product to be used or purchased because of this Environmental Research Laboratories publication.

TABLE OF CONTENTS

FOREWORD

This subject of study serves an important and so far missing link in the planning of GATE research flight missions. The need for optimization of multi-aircraft missions, of flight controlling such missions in view of the possibilities of various on the spot mission changes, and finally the need for scanning rapidly through a number of possible corrective actions in case of emergency moved us to develop a simple computer program to aid in GATE mission planning.

Helmut K. Weickmann

Dr. Helmut K. Weickmann
Director, APCL

COMPUTER SIMULTATION OF MULTI-AIRCRAFT FLIGHTS: AN AID TO MISSION PLANNING

Stephen D. Whitaker

To aid advanced planning in multi-aircraft research flights, a computer program was developed in APCL to simulate a multi-aircraft flight. With this program a scaled, graphic display of each aircraft's position can be obtained at any desired time into the mission. During the development of the program we emphasized flexibility so that flight parameters can be readily changed and the simulated flight re-run for comparison. The program, also, incorporates certain computational aids to provide for aircraft rendezvous and takeoff timing. This report contains a description of the program elements and their operation.

1. INTRODUCTION

The significance of aircraft platforms in atmospheric science research has been enhanced considerably in recent years since the aircraft no longer is just a means of transportation for the scientist and a measuring platform, but rather the aircraft itself has become a sensor of meteorological parameters. As the size of the experimental area increases, so does the necessity for multi-aircraft missions. Investigations using many aircraft simultaneously allow a more thorough coverage of a large position of the system being examined. As the number of aircraft in a mission increases, the necessity for pre-flight and in-flight planning increases.

2. COMPUTER SIMULATION AS AN AID TO PLANNING

Three primary factors are considered in multi-craft mission planning. First is the necessity for efficient use of in-flight time. This becomes more complex as the individual aircraft capabilities vary, i.e., their speed and range. Second is the necessity of coordinating the activities of the aircraft taking part in the mission. This becomes more important when specific aircraft flight configurations, such as a vertical stack, are required. Third is the necessity of rapidly altering preflight and in-flight plans while preserving efficiency, and the goals of the missions.

With these factors in mind, we wrote a computer simulation program. The main objective was to provide a graphic, scaled representation of the position of each aircraft at a specific time.

This program facilitates mission planning in several ways. Given the flight path and speed of each aircraft, the program will compute takeoff intervals to (1) maximize the time in which all aircraft are simultaneously in the appropriate area, or (2) accomplish a rendezvous at some point. Computer simultation allows advance preparation of "canned" flight missions and alternatives. In addition, the capabilities exist for demonstrating alternatives for changing the flight plan after the mission has begun. Such flight alterations may be due to no takeoff of an aircraft; unexpected return of an aircraft; interruption of mission due to special soundings for cloud physics, radiation, or convective phenomenon; unforeseen flight alterations; and emergencies.

3. EXAMPLE OF PROGRAM OUTPUT

The mission in this example involves two aircraft flying the same flight path (fig. 1). Aircraft number 1 flies at 300 knots and aircraft number 2 flies at 200 knots. Figure 2 shows the flight path for both aircraft and the boundary of the area to be investigated.

The aircraft should rendezvous inside the area before flying the "crossing" pattern. Therefore, initially only each path to the rendezvous point is defined (see fig. 3) and the flight time required to reach that

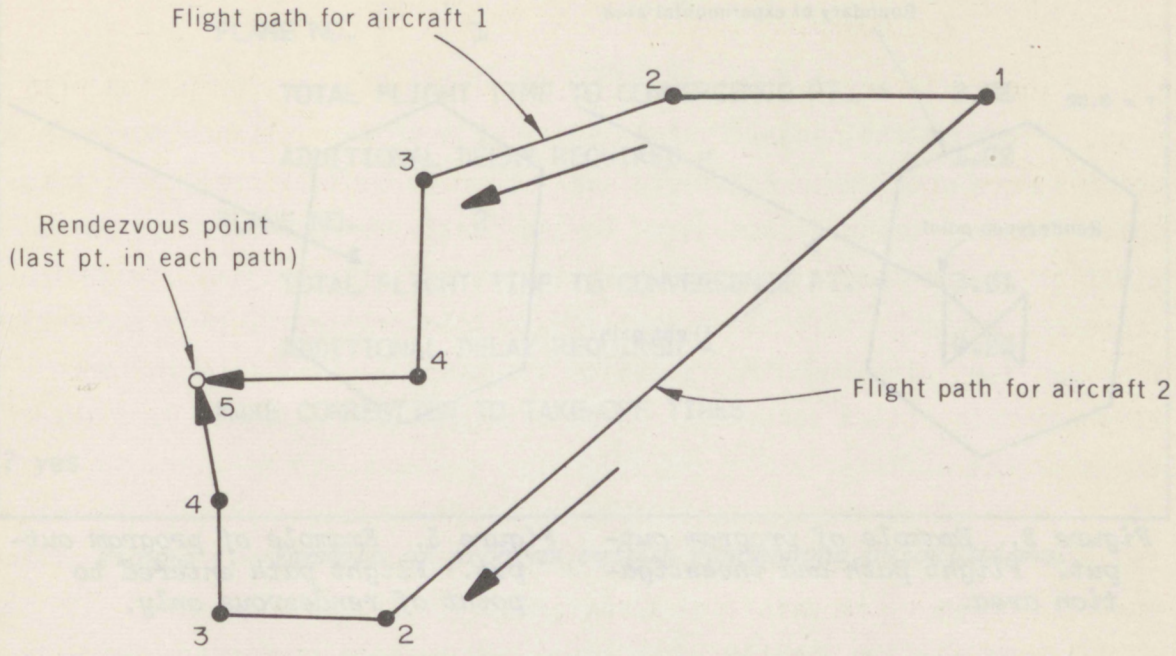

Figure 1. Flight path diagram for two aircraft.

point is calculated (see fig. 4). The takeoff time of aircraft number 1 is delayed 1.00 hour to rendezvous with aircraft number 2 at the second path point.[1]

The remainder of the flight path is added on so as to resemble figure 1 and its total length is calculated and displayed in figure 5.

Figures 6 through 11 show the planes' positions as a function of elapsed time (upper left corner). Their positions, 1 and 2 on the path, correspond to aircraft numbers 1 and 2.

Figure 12 shows two independent flight paths with aircraft positions displayed at 0.25 hour intervals in a single representation. Figure 13 is a sample output for a calculation of the distance between two points in the flight path of figure 2.

---

[1] Path points are defined as those points which, when connected by line segments, define the flight path of the aircraft. They are numbered in the order in which they are encountered by the aircraft. See figure 1.

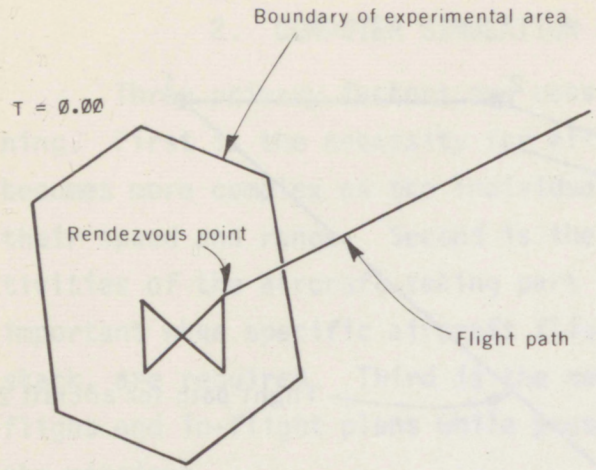

Figure 2. Example of program output. Flight path and investigation area.

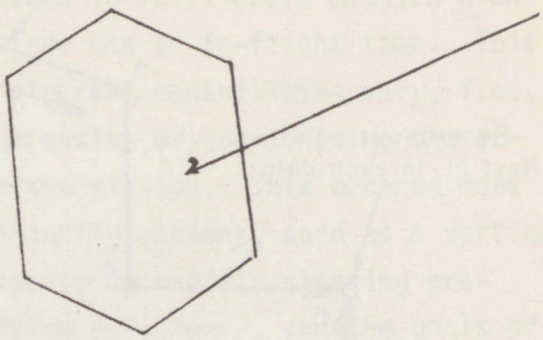

Figure 3. Example of program output. Flight path entered to point of rendezvous only.

4. INPUT-OUTPUT STRUCTURE

4.1. Input Data

1. Points to be connected by the line segments defining the area of investigation.
2. Points to be connected by line segments defining the flight path of each aircraft.
3. Speed and speed changes at points along each aircraft flight path.
4. Time into mission for display of aircrafts' positions.

4.2. Output

1. Take-off timing required for a rendezvous at a common point for two or more aircraft.
2. Flight time for each aircraft to reach rendezvous point.
3. Total length in nautical miles of each flight path.

```
          PLANE NO.      1

              TOTAL FLIGHT TIME TO CONVERGENCE PT. =    2.Ø1

              ADDITIONAL DELAY REQUIRED =               1.ØØ

          PLANE NO.      2

              TOTAL FLIGHT TIME TO CONVERGENCE PT. =    3.Ø1

              ADDITIONAL DELAY REQUIRED =               Ø.ØØ

          MAKE CORRECTION TO TAKE-OFF TIMES
? yes
```

Figure 4. Example of program output rendezvous calculations.

4. Distance between any two points of the flight path.
5. Graphic scaled display of investigation area and individual flight paths.
6. Graphic display for a given time of the position of each aircraft along its flight path.

5. COMPUTER SYSTEM REQUIREMENTS

This program was written for, though not limited to, an on-line computer operation. In this way, the user can achieve the maximum flexibility of the program. Input is through a computer linked graphics tablet and keyboard presently tied to a CDC-6400. The program's memory requirement is approximately 3000 (60-bit) words. The minimum hardware requirements are a teletype terminal and FORTRAN compiler. (The program can also be run completely in batch mode; thus, not even a teletype terminal is necessary.) The software is flexible; being written in FORTRAN IV it is adaptable to any computer system with a FORTRAN compiler. The program source deck has approximately 450 cards.

6. CALCULATION PROCEDURE

6.1. Distance Calculations

The distances between consecutive path points for each aircraft are calculated and stored. Distances over which velocities apply are calculated for each aircraft and added to the total distance from the first point to the one when the particular velocity first applies. These values are then stored. The initial velocity is then integrated over the desired display time. The resulting computed distance is compared with the one appropriate for that velocity. If this computed distance is less than the distance applicable for the velocity, the computed distance is stored. If it is greater, the excess is divided by the initial velocity to obtain the excess time at the point of change; the next velocity is integrated over this time. The total calculated distance is again compared with the distance applicable for the velocity. This process continues until the total computed distance is less than or equal to the applicable distance for the velocity. This final computed distance is the one traveled by the aircraft in the given time.

6.2. Position Calculation

The distance traveled by the aircraft is used to find which two path points bound the aircraft's position. This is done by subtracting the distance between successive path points from the distance traveled until the result is less than or equal to zero.

```
        FLIGHT PATH LENGTH FOR PLANE  1      1773.1 NM

        PATH FOR ANOTHER PLANE   YES, NO, OR SAME

?
```

Figure 5. Example of program output. Calculation of total length of flight path.

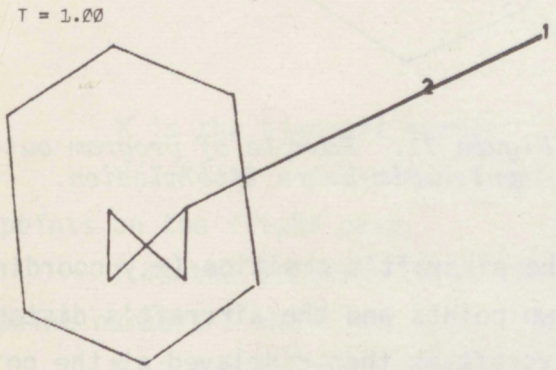

Figure 6. Example of program output. One hour into mission.

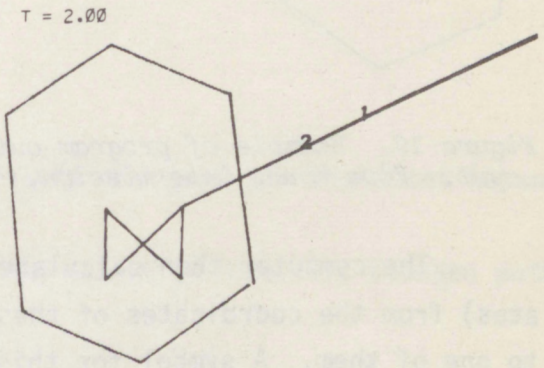

Figure 7. Example of program output. Two hours into mission.

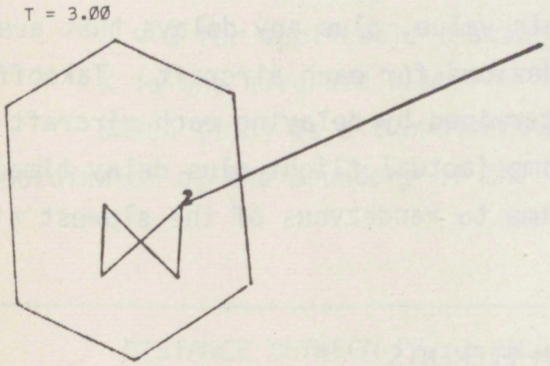

Figure 8. Example of program output. Three hours into mission.

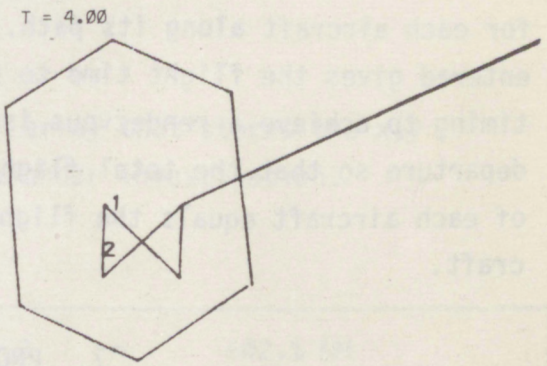

Figure 9. Example of program output. Four hours into mission.

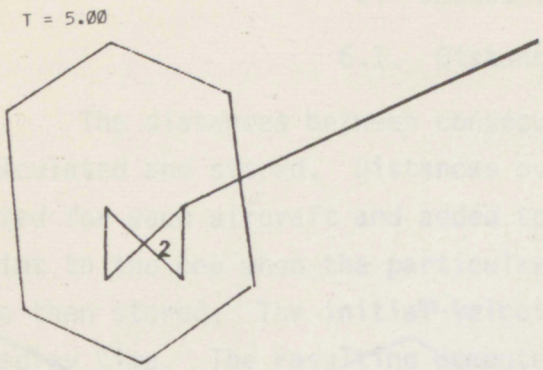

Figure 10. Example of program output. Five hours into mission.

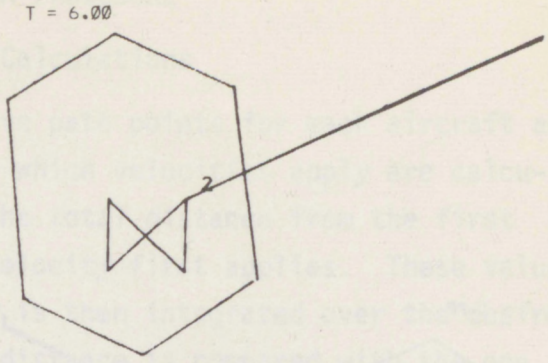

Figure 11. Example of program output. Six hours into mission.

The computer then calculates the aircraft's position (x,y coordinates) from the coordinates of these two points and the aircraft's distance to one of them. A symbol for this aircraft is then displayed at the point x_o,y_o on output. Figure 14 diagrams the procedure.

6.3. Rendezvous and Takeoff Calculations

Rendezvous times are calculated by dividing the distance between path points by the corresponding velocity to determine the time between points. The time between points is summed to the point of the rendezvous for each aircraft along its path. This value, plus any delays that are entered gives the flight time to rendezvous for each aircraft. Takeoff timing to achieve a rendezvous is determined by delaying each aircraft departure so that the total flight time (actual flight plus delay time) of each aircraft equals the flight time to rendezvous of the slowest aircraft.

7. PROGRAM ELEMENTS

7.1. Common Named Variables

PATH (I,J,K) is a three-dimensional array that stores the x,y coordinates of the points defining the path for each aircraft.

I is the path point number

J = 1 for x-coordinates, J = 2 for y-coordinates

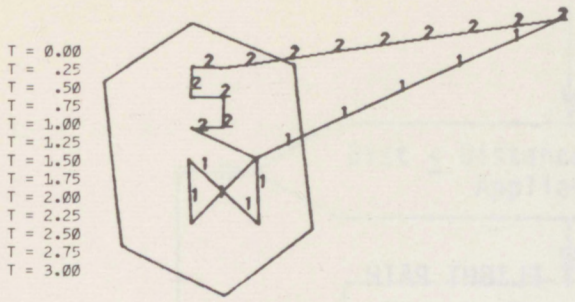

Figure 12. Example of program output. Independent flight paths positions displayed at 0.25 hr intervals.

K is the aircraft number

S(I,K) is a two-dimensional array that stores the distances between points on the flight path.

I is the distance identifier. The stored distance is between path point numbers I and I + 1.

K is the aircraft number

DELTA (I,J,K) is a three-dimensional array that stores the velocity, delay time, and applicable distances for each aircraft. The first two parameters are input; the distances are calculated.

I is the change number

J = 1 for delay time

J = 2 for velocity

J = 3 for applicable distance

K is the aircraft number

BAREA (I,J) is a two-dimensional array that stores the x,y coordinates of the boundary of the area under investigation.

```
        DISTANCE BETWEEN PT  1 AND PT  2     =      602.1 NM

        REPEAT   YES OR NO

?
```

Figure 13. Example of program output. Calculation of distance between two path points.

EACH AIRCRAFT FLIGHT PATH

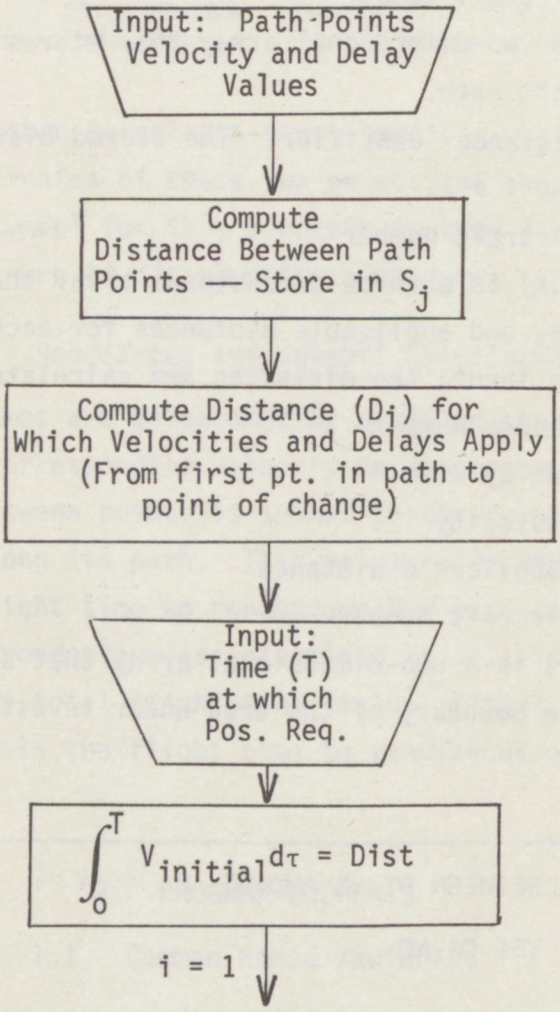

Figure 14. Flow diagram of computations.

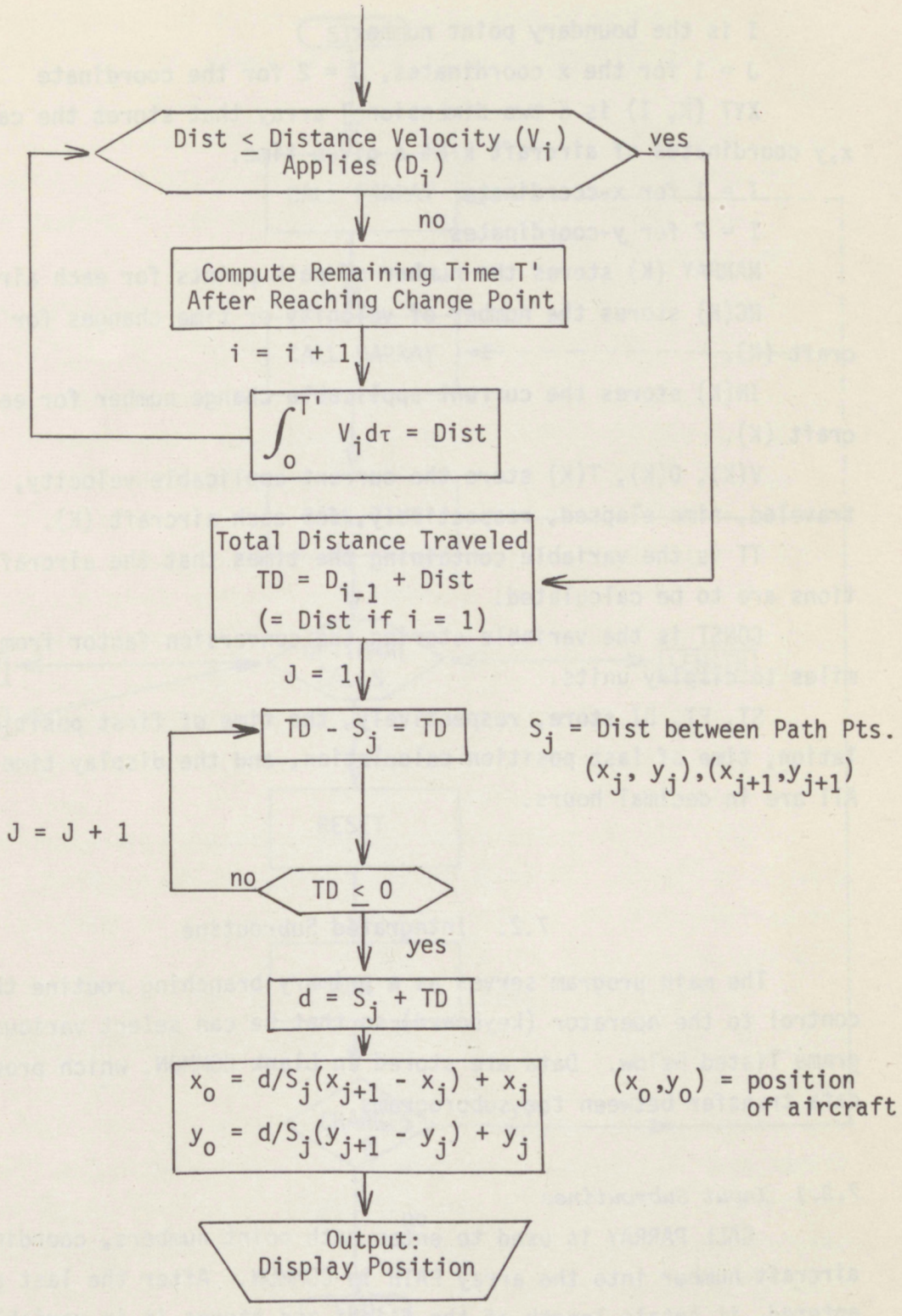

Figure 14. Continued.

I is the boundary point number

J = 1 for the x coordinates, J = 2 for the coordinate

XYT (K, I) is a two-dimensional array that stores the calculated x,y coordinates of aircraft k at a given time.

I = 1 for x-coordinate

I = 2 for y-coordinates

NARRAY (K) stores the number of path points for each aircraft (K).

NC(K) stores the number of velocity or time changes for each aircraft (K).

IN(K) stores the current applicable change number for each aircraft (K).

V(K), D(K), T(K) store the current applicable velocity, distance traveled, time elapsed, respectively, for each aircraft (K).

TT is the variable containing the times that the aircrafts' positions are to be calculated.

CONST is the variable storing the conversion factor from nautical miles to display units.

ST, ET, DI store, respectively, the time of first position calculation, time of last position calculation, and the display time interval. All are in decimal hours.

7.2. Integrated Subroutine

The main program serves as a primary branching routine that returns control to the operator (keyboard) so that he can select various subprograms listed below. Data are stored in blank COMMON, which provides the data transfer between the subprograms.

7.2.1 Input Subroutines

CALL PARRAY is used to enter path point numbers, coordinates, and aircraft number into the array PATH in COMMON. After the last point is entered, it totals length of the flight and stores it in variable XL. This subroutine calls subroutine FLIGHT(K).

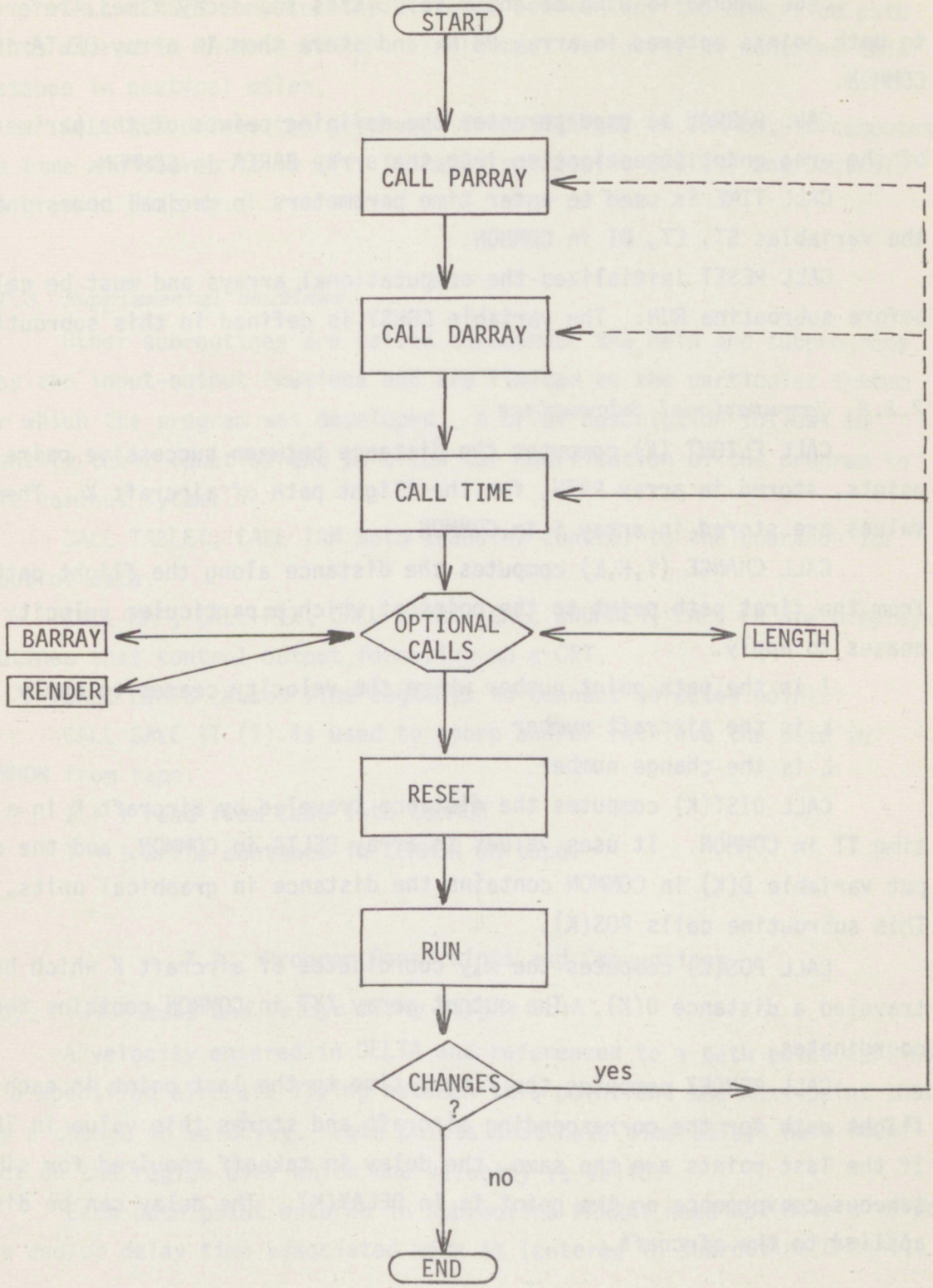

Figure 15. Flow diagram of program operation.

CALL DARRAY is used to enter velocities and decay times, referenced to path points entered in array PATH, and store them in array DELTA in COMMON.

CALL BARRAY is used to enter the defining points of the perimeter of the area under investigation into the array BAREA in COMMON.

CALL TIME is used to enter time parameters in decimal hours into the variables ST, ET, DI in COMMON.

CALL RESET initializes the computational arrays and must be called before subroutine RUN. The variable CONST is defined in this subroutine.

7.2.2 Computational Subroutines

CALL FLIGHT (K) computes the distance between successive pairs of points, stored in array PATH, for the flight path of aircraft K. These values are stored in array S in COMMON.

CALL CHANGE (I,K,L) computes the distance along the flight path from the first path point to the point at which a particular velocity ceases to apply.

I is the path point number where the velocity ceases to apply

K is the aircraft number

L is the change number

CALL DIST(K) computes the distance traveled by aircraft K in a time TT in COMMON. It uses values in array DELTA in COMMON, and the output variable D(K) in COMMON contains the distance in graphical units. This subroutine calls POS(K).

CALL POS(K) computes the x,y coordinates of aircraft K which has traveled a distance D(K). The output array XYT in COMMON contains the coordinates.

CALL RENDEZ computes the flight time to the last point in each flight path for the corresponding aircraft and stores this value in TR(K). If the last points are the same, the delay in takeoff required for simultaneous convergence on the point is in DELAY(K). The delay can be directly applied to the aircraft.

CALL LENGTH computes the distance between any two specified path points stored in PATH in COMMON. The output variable, x, contains this distance in nautical miles.

CALL RUN drives the program. From ST,ET,DI in COMMON, it computes the time and stores it in TT. It calls subroutine DIST(K) and controls the graphic display.

7.2.3 Supplemental Routines

Other subroutines are called throughout the main and subprograms. They are input-output routines and are limited to the particular system for which the program was developed. A brief description follows to identify their function and to allow for modification of the program to suit various sytems.

CALL TABLET, CALL IAM both transfer control to the operator for input of data.

CALL MAP, CALL FIX, CALL PRINT, CALL XNUMBER, CALL ER are display-routines that control output formating on a CRT.

CALL LINES causes line segments to connect selected points.

CALL SAVE IT (I) is used to store and/or retrieve the data in COMMON from tape.

I = 1 read from tape into COMMON

I = 1 write contents in COMMON on tape.

7.3. Program Constraints and Conventions

7.3.1 Velocities and Delays Along Flight Path

A velocity entered in DELTA and referenced to a path point applies to a specified aircraft flying between this point and the next point that has a change in velocity. Path points that have time delays have no effect on the region over which the velocity is valid.

Each path point entered in subroutine PARRAY need not have a velocity and/or delay time associated with it (entered in subroutine DARRAY),

but each change in velocity and/or delay must occur at a path point. (Note: The first path point must have a corresponding velocity entered in subroutine DARRAY.)

To change the value of the velocity and/or delay time at a path point, add or delete a change point (a path point where a change occurs), the entire process of defining change points and values, from first point to last, must be repeated. No isolated changes are possible.

When a time delay is encountered at a charge point, the aircraft's position is held at that point until the display time exceeds the elapsed time plus the delay time.

7.3.2 Path Points

Path points are entered in PATH in the order in which they are encountered along the intended flight path.

7.3.3 Initialization

Initially, a call to subroutine RESET is required before a call to RUN. A call to RESET is required after a call to RUN only if the display time, TT, is less than the previous display time, or if the flight path or velocities and delays times have been altered.

7.3.4 Rendezvous and Takeoff Calculations

The program assumes the last point in each flight path is a common one. If in reality the rendezvous is not the last point in the entire flight path, then only path points up to the rendezvous should be initially entered in PATH. Then this calculation can be carried out and the additional path points added subsequently through subroutine PARRAY.

7.3.5 Array Limits

The number of aircraft, path points, change points, and boundary points are limited by the array dimensions named in COMMON. As written,

five aircraft can be accommodated. There is, however, no restriction on increasing the array's dimensions other than those imposed by the computer system.

7.3.6 Time Conventions

All input and output values for time are in decimal hours. Elapsed time is measured from T = 0.00 when the first aircraft leaves the runway. To display aircraft positions for a single time, the variables ST and ET in COMMON must both be equal to the given time.

7.4. Normal Order of Subroutine Calls

The normal order of calling subroutines is listed below. Figure 15 shows a flow diagram of the subroutines in the program.

CALL RESET — Initialize arrays

CALL PARRAY — Enter path points

CALL DARRAY — Enter change points and values (changes must be entered in order of occurrence and there must be a velocity associated with the first path point).

CALL TIME — Enter beginning time, ending time, and display interval. If beginning time and ending time are the same the aircraft position will be displayed only at that time.

CALL BARRAY (optional) — Enter boundary points for investigation area.

CALL RENDEZ (optional) — Enter aircraft converging.

NOTE: The last point in each flight path is assumed as the rendezvous point. Therefore, if the rendezvous point is actually midway in the flight path, initially enter the path point only as far as the rendezvous point and compute the times, then add the remainder of the path by calling PARRAY.

CALL LENGTH (optional) — Distance between points.

CALL RUN — Calculates position and drives display routines.

CALL RESET — Before another call to RUN.

8. ACKNOWLEDGMENT

Elemer Magaziner of APCL provided invaluable advice on the CDC-6400 input-output structure with respect to the CRT and Tablet. In addition, he made many worthwhile suggestions concerning the content of the program itself.

```
      PROGRAM FLISIM(INPUT,OUTPUT,TAPE1)
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      DIMENSION IARR(8)
      CALL SAVEIT(0)
      CALL MAP(0.,1023.,0.,800.,0.,1.0,0.,0.782)
      CALL RESET
      GO TO 2
   15 CALL ER
      ITYPE=5
      CALL FIX(100,600)
      PRINT,*NEXT CHOICE, ENTER LIST FOR LIST OF OPTIONS*
      CALL IAM(0,0,1,IARR,ITYPE,0,0)
      IF(ITYPE.EQ.2) GO TO 11
    2 CALL ER
      CALL FIX(100,600)
      PRINT,*PATH POINTS   ENTER 1*
      CALL FIX(100,550)
      PRINT,*CHANGE POINTS AND VALUES   ENTER 2*
      CALL FIX(100,500)
      PRINT,*NEW TIME PARAMETERS   ENTER 3*
      CALL FIX(100,450)
      PRINT,*RUN   ENTER 4*
      CALL CSTRING(100,400,22LINITIALIZE   ENTER 5$.)
      CALL CSTRING(100,350,54LCOMPUTE TIME TO RENDEZVOUS AND T-O SPACING
     1   ENTER 6$.)
      CALL CSTRING(100,300,25LB-AREA POINTS   ENTER 7$.)
      CALL CSTRING(100,250,33LFINISHED WITH PROGRAM   ENTER 8$.)
      CALL CSTRING(100,200,55LTO DETERMINE DISTANCES BETWEEN TWO PATH PT
     1S   ENTER 9$.)
      CALL IAM(0,0,1,IR,2,0,0)
      GO TO 12
   11 IR=IARR(1)
   12 GO TO (3,4,5,6,1,7,8,9,10),IR
    1 CALL RESET
      GO TO 15
    3 CALL PARRAY
      CALL SAVEIT(1)
      GO TO 15
    4 CALL DARRAY
      CALL SAVEIT(1)
      GO TO 15
    5 CALL TIME
      GO TO 15
    6 CALL RUN
      GO TO 15
```

```
  7 CALL RENDEZ
    CALL SAVEIT(1)
    GO TO 15
  8 CALL BARRAY
    CALL SAVEIT(1)
    GO TO 15
 10 CALL LENGTH
    GO TO 15
  9 CALL SAVEIT(1)
    END


    SUBROUTINE FLIGHT(K)
    COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
   1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
    SUM =0.0
    MARRAY=NARRAY(K)-1
    DO 25 I=1,MARRAY
    P1=PATH(I,1,K)-PATH(I+1,1,K)
    P2=PATH(I,2,K)-PATH(I+1,2,K)
    S(I,K)=SQRT(P1*P1+P2*P2)
 25 SUM=SUM+S(I,K)
    S(MARRAY+1,K)=SUM
    RETURN
    END


    SUBROUTINE POS(K)
    COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
   1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
    I=0
    B=D(K)
    IF(B.LE.0.0) GO TO 120
105 IF(I+1.EQ.NARRAY(K)) GO TO 130
    I=I+1
    D1=B
    B=D1-S(I,K)
    IF(B) 110,115,105
110 P1=PATH(I+1,2,K)-PATH(I,2,K)
    P2=PATH(I+1,1,K)-PATH(I,1,K)
    XYT(K,1)=D1/S(I,K)*P2+PATH(I,1,K)
    XYT(K,2)=D1/S(I,K)*P1+PATH(I,2,K)
    GO TO 150
115 XYT(K,1)=PATH(I+1,1,K)
    XYT(K,2)=PATH(I+1,2,K)
    GO TO 150
120 XYT(K,1)=PATH(1,1,K)
    XYT(K,2)=PATH(1,2,K)
    GO TO 150
```

```
130 M=NARRAY(K)
    XYT(K,1)=PATH(M,1,K)
    XYT(K,1)=PATH(M,2,K)
150 CONTINUE
    RETURN
    END

    SUBROUTINE BARRAY
    DIMENSION ISW(2)
    COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
   1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
    CALL ER
    CALL FIX(100,600)
    PRINT,*ENTER BOUNDARY PTS. FOR B-AREA*
    L=0
  1 CALL TABLET(IX,IY,IPR,IPP,IPB,ISW)
    IF(ISW(1).EQ.0) L=L+1
    BAREA(L,1)=IX
    BAREA(L,2)=IY
    IF(L.LT.7) GO TO 1
    RETURN
    END


    SUBROUTINE DIST(K)
    COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
   1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
    T1=TT
    I=IN(K)
    IF(T(K).EQ.0.0) GO TO 2
  1 IF (T1.LE.T(K)) GO TO 7
    DD =D(K)
    D(K)=V(K)*(T1-T(K))+D(K)
    IF (DELTA(I,3,K).GT.D(K)) GO TO 4
    D(K)=DELTA(I,3,K)
    T(K)=T(K)+(DELTA(I,3,K)-DD)/V(K)
    I=I+1
    IF (I.GT.NC(K)) GO TO 8
  2 V(K)=DELTA(I,2,K)
  3 T(K)=T(K)+DELTA(I,1,K)
    GO TO 1
  4 T(K)=T(K)+(D(K)-DD)/V(K)
  7 IN(K)=I
    CALL POS(K)
    GO TO 10
  8 A=K*25.
    CALL FIX(100,100)
    PRINT,*FLIGHT PARAMETERS HAVE BEEN EXCEEDED FOR PLANE*
    CALL XNUMBR(660.+A,100.,K,2HI1)
 10 RETURN
    END
```

```
      SUBROUTINE RESET
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      CONST=7./12.
      DO 5 I=1,5
      D(I)=0.0
      T(I)=0.0
      IN(I)=1
    5 CONTINUE
      RETURN
      END


      SUBROUTINE PARRAY
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      DIMENSION ISW(2),ZZ(8),IARR(2),ITYPE(2)
      DATA ITYPE/2*2/
    4 CALL ER
      CALL FIX(100,600)
      PRINT,*ENTER PLANE NO. AND BEGINNING PATH PT NO.*
      CALL IAM(0,0,2,IARR,ITYPE,0,0)
      K=IARR(1)
      I=IARR(2)-1
    1 CALL TABLET(IX,IY,IPR,IPP,IPB,ISW)
      IF(ISW(1).EQ.0) I=I+1
    2 PATH(I,1,K)=IX
      PATH(I,2,K)=IY
      IF(ISW(2).EQ.0) GO TO 1
      NARRAY(K)=I
      CALL FLIGHT(K)
      CALL ER
      CALL FIX(100,600)
      PRINT,*FLIGHT PATH LENGTH FOR PLANE*
      CALL XNUMBR(470.,600.,K,2HI1)
      XL=S(I,K)/CONST
      CALL XNUMBR(500.,600.,XL,9HF6.1,2HNM)
    3 CALL FIX(100,550)
      PRINT,*PATH FOR ANOTHER PLANE   YES, NO, OR SAME*
      CALL IAM(0,0,1,ZZ,4,0,0)
      IF(ZZ.EQ.3HYES) GO TO 4
      IF(ZZ.EQ.2HNO) GO TO 5
      CALL ER
      CALL FIX(100,500)
      PRINT,*SAME PATH FOR WHICH PLANE*
      CALL IAM (0,0,1,K1,2,0,0)
      DO 10 M=1,I
      PATH(M,1,K1)=PATH(M,1,K)
      PATH(M,2,K1)=PATH(M,2,K)
```

```
10 S(M,K1)=S(M,K)
   NARRAY(K1)=I
   CALL ER
   GO TO 3
 5 RETURN
   END

   SUBROUTINE DARRAY
   COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
  1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
   DIMENSION IK(8),IX(8),IA(2),ITYPE(3),ITY(2),AB(3)
   DATA ITYPE /1,1,5/
 4 CALL ER
   L=0
   CALL FIX(100,600)
   PRINT,*ENTER PLANE NO.*
   CALL IAM(0,0,1,K,2,0,0)
 1 CALL ER
   IK=0
   L=L+1
   CALL FIX(100,700)
   PRINT,*ENTER PATH POINT NO.*
   CALL FIX(100,600)
   PRINT,*ENTER DELAY TIME IN DECIMAL HRS.*
 2 CALL FIX(100,500)
   PRINT,*ENTER NEW SPEED IN DECIMAL KNOTS, OR SAME*
   CALL IAM(0,0,3,AB,ITYPE,0,0)
   I=AB(1)
   IF(ITYPE(3).EQ.4) GO TO 10
   DELTA(L,2,K)=AB(3)*CONST
   GO TO 11
10 DELTA(L,2,K)=DELTA(L-1,2,K)
11 DELTA(L,1,K)=AB(2)
 6 CALL ER
   CALL FIX(100,400)
   PRINT,*FOR ANOTHER PT. OF CHANGE ENTER YES OR NO*
   CALL IAM(0,0,1,IK,4,0,0)
   IF(IK.EQ.2HNO) GO TO 3
   IF(L.EQ.1) GO TO 1
   CALL CHANGE(I,K,L-1)
   GO TO 1
 3 IF(L.NE.1) CALL CHANGE(I,K,L-1)
   NA=NARRAY(K)
   DELTA(L,3,K)=S(NA,K)
   NC(K)=L
   CALL ER
   CALL FIX(100,600)
   PRINT,*ENTER CHANGES FOR ANOTHER PLANE   YES OR NO*
   CALL IAM(0,0,1,IX,4,0,0)
   IF(IX.EQ.3HYES) GO TO 4
   RETURN
   END
```

```
      SUBROUTINE CHANGE(I,K,L)
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      SUM=0.0
      N=I-1
      DO 5 J=1,N
    5 SUM=SUM+S(J,K)
      DELTA(L,3,K)=SUM
      RETURN
      END



      SUBROUTINE RENDEZ
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      DIMENSION IM(5),DELAY(5),TR(5),ITYPE(5)
      DIMENSION NP(8)
      DATA ITYPE/5*2/
      INTEGER A
      CALL ER
      CALL FIX(100,600)
      PRINT,*ENTER NO. OF PLANES CONVERGING*
      CALL IAM(0,0,1,N,2,0,0)
      CALL ER
      CALL FIX(100,500)
      PRINT,*ENTER PLANE NUMBERS*
      CALL IAM(0,0,N,IM,ITYPE,0,0)
      II=0.0
      DO 4 M=1,N
      K=IM(M)
      TE=0.0
      DD=0.0
      DT=0.0
      NB=NC(K)
      VV=DELTA(1,2,K)
      DT=DELTA(1,1,K)
      IF(NB.EQ.1) GO TO 5
      DO 3 I=2,NB
      TE=(DELTA(I-1,3,K)-DD)/VV+TE
      DD=DELTA(I-1,3,K)
      VV=DELTA(I,2,K)
    2 DT=DT+DELTA(I,1,K)
    3 CONTINUE
      GO TO 6
    5 TE=(DELTA(1,3,K)-DD)/VV+TE
    6 TR(K)=TE+DT
      IF(TR(K).GT.T1) T1=TR(K)
```

```
 4 CONTINUE
   CALL ER
   A=0
   DO 10 M=1,N
   K=IM(M)
   CALL FIX(100,750-A)
   PRINT,*PLANE NO.*
   CALL XNUMBR(300.,750.-A,K,2HI1)
   CALL FIX(150,700-A)
   PRINT,*TOTAL FLIGHT TIME TO CONVERGENCE PT. =*
   CALL XNUMBR(700.,700.-A,TR(K),4HF6.2)
   CALL FIX(150,650-A)
   PRINT,*ADDITIONAL DELAY REQUIRED =*
   DELAY(K)=T1-TR(K)
   CALL XNUMBR (700.,650.-A,DELAY(K),4HF6.2)
   A=A+150
10 CONTINUE
   CALL FIX(100,25)
   PRINT,*MAKE CORRECTION TO TAKE-OFF TIMES *
   CALL IAM(0,0,1,NP,4,0,0)
   IF(NP.EQ.2HNO) GO TO 15
   DO 12 M=1,N
   K=IM(M)
12 DELTA(1,1,K)=DELTA(1,1,K)+DELAY(K)
15 RETURN
   END


   SUBROUTINE TIME
   COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
  1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
   DIMENSION ARR(3),ITYPE(3)
   DATA ITYPE/3*1/
   CALL ER
   CALL FIX(100,600)
   PRINT,*ENTER START TIME IN DECIMAL HRS.*
   CALL FIX (100,500)
   PRINT,*ENTER END TIME IN DECIMAL HRS.*
   CALL FIX(100,400)
   PRINT,*ENTER DISPLAY INTERVAL*
   CALL IAM(0,0,3,ARR,ITYPE,0.0)
   ST=ARR(1)
   ET=ARR(2)
   DI=ARR(3)
   RETURN
   END
```

```
      SUBROUTINE LENGTH
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      DIMENSION IA(3), ITYPE(3), AB(8)
      DATA ITYPE/3*2/
    1 CALL ER
      CALL FIX(100,600)
      PRINT,*ENTER PLANE NO., STARTING PATH PT NO., ENDING PATH PT NO.*
      CALL IAM(0,0,3,IA,ITYPE,0,0)
      CALL ER
      CALL FIX(100,600)
      PRINT,*DISTANCE BETWEEN PT    AND PT    =*
      CALL XNUMBR(350.,600.,IA(2),2HI1)
      CALL XNUMBR(460.,600.,IA(3),2HI1)
      X=XLENG(IA(2),IA(3),IA(1))/CONST
      CALL XNUMBR(580.,600.,X,9HF6.1,2HNM)
      CALL FIX(100,550)
      PRINT,*REPEAT   YES OR NO*
      CALL IAM(0,0,1,AB,4,0,0)
      IF(AB.EQ.3HYES) GO TO 1
      RETURN
      END


      FUNCTION XLENG(I,J,K)
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      JJ=J-1
      XLENG=0.0
      DO 1 N=I,JJ
    1 XLENG=XLENG+S(N,K)
      RETURN
      END


      SUBROUTINE RUN
      COMMON   PATH(25,2,5),S(25,5),XYT(5,2),NARRAY(5),DELTA(20,3,5),
     1NC(5),IN(5),V(5),D(5),T(5),TT,CONST,BAREA(7,2),ST,ET,DI
      DIMENSION KK(5),IARR(8),ITYPE(5)
      DATA ITYPE/5*2/
      TT=ST
      CALL ER
      CALL FIX(100,600)
      PRINT,*ENTER NO. OF PLANES*
      CALL IAM(0,0,1,N,2,0,0)
      CALL ER
      CALL FIX(100,500)
      PRINT,*ENTER PLANE NOS.*
      CALL IAM(0,0,N,KK,ITYPE,0,0)
      CALL ER
```

```
      CALL LINES(BAREA(1,1),BAREA(1,2),7)
      DO 11 NZ=1,N
      K=KK(NZ)
   11 CALL LINES(PATH(1,1,K),PATH(1,2,K),NARRAY(K))
      AY=750.
    1 CALL XNUMBR (50.,AY,TT,9H2HT=,F6.2)
      DO 2 NK=1,N
      K=KK(NK)
      CALL DIST(K)
      CALL XNUMBR (XYT(K,1),XYT(K,2),K,2HI1)
    2 CONTINUE
      AY=AY-25.
      IF(TT.GE.ET) GO TO 3
      TT=TT+DI
      GO TO 1
    3 CALL IAM(0,0,1,IARR,4,0,0)
      RETURN
      END


      SUBROUTINE FIX (IX,IY)
      CALL MA(IX,IY)
      CALL MODE(0)
      CALL FLSH
      RETURN
      END



      SUBROUTINE SAVEIT(INDEX)
      COMMON A(734)
      JNDEX=INDEX+1
      GO TO (1,2),JNDEX
    1 CALL GET(1,4HGATE,0,0)
      REWIND 1
      READ(1) (A(I),I=1,734)
      RETURN
    2 REWIND 1
      WRITE(1) (A(I),I=1,734)
      ENDFILE 1
      CALL REPLACE(1,4HGATE,0,0)
      RETURN
      END
```


☆GPO 1973 — 784-570 REGION NO. 8