NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE
EVALUATING AND MINIMIZING NOISE IMPACT DUE TO AIRCRAFT FLYOVER

A Final Report

Submitted to:
NASA Scientific and Technical Information Facility
P. O. Box 8757
Baltimore/Washington International Airport
Baltimore, Maryland 21240

Submitted by:
Ira D. Jacobson
Professor
Gerald Cook
Professor

Report No. UVA/528166/MAE80/102
May 1980
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES

Members of the faculty who teach at the undergraduate and graduate levels and a number of professional engineers and scientists whose primary activity is research generate and conduct the investigations that make up the school's research program. The School of Engineering and Applied Science of the University of Virginia believes that research goes hand in hand with teaching. Early in the development of its graduate training program, the School recognized that men and women engaged in research should be as free as possible of the administrative duties involved in sponsored research. In 1959, therefore, the Research Laboratories for the Engineering Sciences (RLES) was established and assigned the administrative responsibility for such research within the School.

The director of RLES—himself a faculty member and researcher—maintains familiarity with the support requirements of the research under way. He is aided by an Academic Advisory Committee made up of a faculty representative from each academic department of the School. This Committee serves to inform RLES of the needs and perspectives of the research program.

In addition to administrative support, RLES is charged with providing certain technical assistance. Because it is not practical for each department to become self-sufficient in all phases of the supporting technology essential to present-day research, RLES makes services available through the following support groups: Machine Shop, Instrumentation, Facilities Services, Publications (including photographic facilities), and Computer Terminal Maintenance.
A Final Report

EVALUATING AND MINIMIZING NOISE IMPACT DUE TO AIRCRAFT FLYOVER

Submitted to:
NASA Scientific and Technical Information Facility
P. O. Box 8757
Baltimore/Washington International Airport
Baltimore, Maryland 21240

Submitted by:
Ira D. Jacobson
Professor
Gerald Cook
Professor

Department of Mechanical and Aerospace Engineering
RESEARCH LABORATORIES FOR THE ENGINEERING SCIENCES
SCHOOL OF ENGINEERING AND APPLIED SCIENCE
UNIVERSITY OF VIRGINIA
CHARLOTTESVILLE, VIRGINIA

Report No. UVA/528166/MAE80/102
May 1980
I. INTRODUCTION

Contained in this report are the results of a study on the evaluation and reduction of noise impact to a community due to aircraft landing and takeoff operations. This work is a continuation of the methods and results of a previous study done by the same authors (under NASA Grant NSG-1509, reference 1). For completeness some repetition of the earlier work is included.

The previous work considered only a single aircraft using a single approach/landing trajectory. Models of population distribution, aircraft noise signature, and aircraft flight path were developed, and a suitable annoyance model adopted. A performance index to be minimized was formed from the annoyance model and constraints. The current study has examined the case of multiple aircraft, flying on several trajectories, for either the case of approach/landings or for takeoffs. A superior, more realistic model of the flight path has also been developed. As in the earlier work, the annoyance criterion used is the noise impact index (NII). The algorithm developed has been applied to Patrick Henry International Airport.

Discussions of the various models, the performance index, optimization methods, and results appear in the following sections.
II. PROBLEM FORMULATION

OVERVIEW

The problem considered is that of determining the "best" set of aircraft landing and/or takeoff paths from any airport which minimizes the noise impact on the surrounding community. There are five major aspects of this problem which must be modeled: (1) aircraft noise signatures, (2) population distributions, (3) a cost function or performance index, (4) the aircraft flight paths, and (5) constraints on the aircraft (based upon aircraft dynamics), passenger comfort, safety, and maximum noise exposure for any population group. In addition, a flight path optimization scheme must be adopted. A modular concept has been employed so that any section of the problem may be modified with relative ease. The following sections describe each of these in detail.

A. Aircraft Noise Signatures

An aircraft noise signature gives a description of the noise emanating from an aircraft. Many such representations are available. The one adopted here is a simple model to facilitate computation; however, it can be replaced with more complex and accurate models. One such model is available through the use of the Aircraft Noise Source and Contour Estimation computer programs (see references 2, 3). The aircraft noise signature used in this study is obtained using data from reference 4. Here the effective perceived noise level (EPNdB) is given as a function of slant range to the closest point of approach for a variety of aircraft. A typical plot of the slant range variation for two different aircraft is shown in Figure 1. These data were fit using standard least squares techniques to yield an expression for EPNdB given by
Figure 1. EPNdB vs. Slant Range

FLYBY NOISE LEVEL
(1.93 - 1.95 EPR 727 Aircraft) FIG. A-1**
(1.94 EPR  DC-9 Aircraft) FIG. D-1**

*Closest Point of Approach
**FAA-RD-71-83 (Ref. 6)
EMB = 115 - 22.5 \times \left( \frac{\text{slant range in ft.}}{500} \right)

This equation is used for calculation of the maximum noise level at each location for a flyover. A typical footprint for a straight-in approach along a 3-degree glide slope is shown in Figure 2. For other aircraft, similar experimental data must be sought in the literature.

B. Population Distribution Model

To model the distribution of population, a map of the community is overlaid with a grid and the population in each grid section is determined. The population distribution within each section is assumed to be uniform. Several grid geometries were examined (see Figure 3). The geometries include: (1) rectangular sections of equal size, (2) rectangular sections whose dimensions increase with distance from the airport runway, and (3) concentric circles divided by several radial lines. The second scheme was chosen since it requires fewer rectangular sections than the first and is somewhat easier to implement than the third. Computer time required for determining the optimum trajectory varies directly with the number of grid sections. This results in the desire to minimize the number of blocks in the grid. Furthermore, since the noise levels decrease with distance from the aircraft and the aircraft has higher altitude when farther from the runway, the need for high resolution of the population density diminishes with distance from the airport. Grid blocks with larger area may then be used when farther away from the airport.

Within a grid section, the population is determined by use of the SITE II system (reference 5), available on the CDC 7600 computer at the
Figure 2. Noise Footprint

x,y Coordinates Unit: Mile
A.C. Original Altitude: 20000 Feet
A.C. Speed: 279 ft./sec
3 Degree Landing
Figure 3. Population Grid Geometry

1. Equal Size Blocks
2. Variable Size Blocks
3. Concentric Circles
NASA-Langley facility. This system requires as input the latitude and longitude of a reference point and the coordinates of the corners of each rectangular section. Although SITE II allows for simple retrieval of 1970 census data, there is some question about its resolution capabilities for small grid sections. In addition, in rapidly growing areas the population data may lag the actual population. The SITE II program is capable of producing detailed census information as shown in Figure 4; however, for the present analysis only the population information is used, as indicated.

C. Flight Path Model

There are two ways in which the trajectory of the aircraft may be determined. In one, a discrete time integration of the equations of motion (with control deflections) yields point by point spatial coordinates and orientation. Although this allows the flexibility of explicitly including control constraints as well as dynamic constraints (e.g. maximum roll angle), it requires that a considerable number of states of the system be stored in the optimization routine (i.e. each point of the trajectory in discrete form). In the multi-aircraft, multi-trajectory problem investigated here, such storage requirements are prohibitive.

Thus, another method was adopted which uses only the functional form of the trajectory to describe the flight path. Two possibilities have been investigated: (1) a truncated Fourier series representation and (2) a scheme of line segments joined by smooth arcs.

The Fourier series has the advantage of being able to represent any smooth function over a finite range reasonably well when the series is
Figure 4. Demographic Profile Report from SITE II
truncated after a few terms. However, it is not able to represent functions with slope discontinuities without introducing "waviness" into the approximation. A large number of terms are needed to reduce this effect. The line segment representation does not have either of these features; however, it can approximate very well functions which describe the types of paths aircraft customarily fly.

The first method begins by generating a starting path which goes from the initial trajectory point to the desired runway, ending up with the proper heading, i.e., the aircraft velocity vector is aligned with the runway. This starting trajectory is generated using the following equation: (see Figure 5)

$$y_s(x) = [m_f(x-x_p) + (y_p-y_o)] \exp \left[ -C_1 \left( \frac{x-x_f}{x_o-x_f} \right) \right] + y_o$$

For the vertical motion a simple three-degree descent path was assumed.

Next, the first five Fourier sine harmonics are used to introduce deviations from the starting path. The coordinate system is scaled so that each of the sine functions contributes zero deviation at the end points. Therefore, if the starting path satisfies the boundary conditions, then the path with the deviations will also. An exponentially decaying factor is used to eliminate heading deviations at the final point.

With the deviations, the equations for the path become

$$y = \sum_{i=1}^{N} \alpha_i \sin \left[ i\pi \left( \frac{x-x}{x_f-x_o} \right) \right] \left[ 1 - \exp \left( -\frac{x-x_f}{C_2} \right) \right] + y_s(x)$$

$$z = \sum_{i=1}^{N} \beta_i \sin \left[ i\pi \left( \frac{x-x}{x_f-x_o} \right) \right] \left[ 1 - \exp \left( -\frac{x-x_f}{C_2} \right) \right] + z_s(x)$$
Slope = \frac{y_f - y_p}{x_f - x_p}

Figure 5. Rotated Coordinate System for Establishing Nominal Flight Trajectory from Initial Point to Runway Approach.
where the $\alpha_i$ and $\beta_i$ are the unknowns to be determined.

The second flight path model represents the trajectory as a chain of line segments extending from the initial to the final point. Each corner between two segments is "smoothed" with a circular arc whose radius is large enough to insure that the aircraft can perform the turn (see Figure 6). The unknown variables to be determined are the coordinates of the line segment intersections (corner points). For the starting trajectory, the corner points lie equally spaced along a line through each pair of initial and final points. The number of line segments and hence, the number of corner points, per trajectory is determined before the optimization begins. This number is generally small (3 to 5) so that the pilot is not overburdened with required maneuvers.

Both models of the flight path have the advantage of requiring only a small number of parameters to describe the trajectory. This reduces the optimization problem from a variational one to an ordinary one, but care must be taken to see that the various constraints in the problem are met.

D. Constraints

The use of a functional form of the flight path for the trajectory requires the reformulation of constraints into parameters which can be used in the optimization. This is accomplished by translating the steady state solutions of the lateral and longitudinal perturbation equations into geometric constraints. For a detailed derivation of these, see the final report for 1979, Appendix A of reference 1. The constraints are incorporated by determining maximum curvature and slope parameters as a function of aerodynamic and physical constraints.
Similar expressions are given in the appendix (referred to above) for constraints on aileron, rudder, and elevator deflections, flight path angle and pitch rate limits.

In addition to the aircraft constraints, there are passenger comfort considerations (e.g. max bank angle), maximum noise exposure levels, and a minimum separation distance between multiple trajectories.

All of the constraints are listed below:

I. Aircraft Dynamic Constraint:

\[
\frac{d^2y}{dx^2} \leq \frac{C_1 + C_2 C_3}{V_{\text{avg}}} \min (\delta r_1, \delta r_2, \delta r_3)
\]

where \( V_{\text{avg}} \) = average velocity of aircraft, \( C_1, C_2, C_3 \) are constants for a given aircraft, and the \( \delta r \)'s depend upon maximum bank angle and maximum rudder and aileron deflections for a given aircraft.

Longitudinal:
\[
\tan \gamma_{c_{\text{max}}} \leq \frac{dz}{dx} \leq \tan \gamma_{d_{\text{max}}}
\]

where \( \gamma_{c_{\text{max}}} \) and \( \gamma_{d_{\text{max}}} \) are the maximum climb and descent angles.

II. Passenger Comfort Constraint:

\[
\frac{1 + \left(\frac{dy}{dx}\right)^2}{dx} \leq \frac{d^2y}{dx^2} \leq \frac{V_{\text{avg}}^2}{C_4 g}
\]

where \( C_4 = 1.9 \) for 90% passenger satisfaction, 4.5 for 80% satisfaction, \( g = \) acceleration due to gravity, and \( V_{\text{avg}} \) = average velocity of aircraft during the turn.
III. Threshold Noise Constraint

No populated area may receive noise in excess of 95 dB more than \( N \) percentage of times per day, where \( N \) is a fixed percentage of the number of flights per day. \( N \) is made as small as possible for any given case.

IV. Minimum Separation Constraint

A minimum distance of 800 meters (\( \frac{1}{2} \) mile) must be maintained between any two trajectories at all points (except very close to the runway, where all trajectories must converge).

E. Cost Function

A large number of criteria have been proposed to evaluate noise annoyance (e.g., EPNdB, NNI, sleep interference index, speech interference index, etc.). The recent trend in noise assessment work is toward a universal measure -- the noise impact index (NII). This measure is a weighted day-night model which accounts for population density. It is described in detail in reference 6. Briefly, the total population exposed to each incremental average day-night model sound level is multiplied by the weighting function for that level. The weighting factor \( W(L_{dn}) \), multiplied by the population exposed to that \( L_{dn} \), is summed and normalized by the total population giving the Noise Impact Index for the area:

\[
NII = \frac{\sum_{L_{dn}} P(L_{dn}) W(L_{dn})}{\sum_{L_{dn}} P(L_{dn})}
\]

A plot of \( W(L_{dn}) \) appears in Figure 7.
SOUND LEVEL WEIGHTING FUNCTION
FOR OVERALL IMPACT ANALYSIS

Figure 7. Impact Intensity Weighting Function
The cost function or performance index for the optimization procedure is taken to be the NII plus penalties for violating constraints. Basically, the optimization procedure is set up to "drive" the aircraft trajectories to the path which will minimize the NII and at the same time, not violate any constraints. As an example, the constraint of flight path angle not exceeding a maximum descent angle, $\gamma_d$, nor a maximum climb angle, $\gamma_c$, is written as

$$\tan \gamma_c < \left| \frac{dz}{dx} \right| < \tan \gamma_d$$

Each is converted to a penalty which is added to the NII in the form

$$\text{Cost} = \text{NII} + K_1 P_1 + K_2 P_2$$

$$P_1 = \left\{ \max[0, (\tan \gamma_c - \frac{dz}{dx})] \right\}^2$$

$$P_2 = \left\{ \max[0, (\frac{dz}{dx} - \tan \gamma_d)] \right\}^2$$

$k_1, k_2$ = constants

As is seen, for values of the flight path angle within the allowable range, no penalty is added; however, for values outside this range, the penalty and thus, the increase in cost, is great. Other penalty terms are added in a like manner.
III. OPTIMIZATION

The optimum set of trajectories is determined by calculating values of the unknowns (the \( \alpha_i \) and \( \beta_i \) in the Fourier series model, the corner points in the line segment model) which minimize the total cost (NII plus penalties). Two optimization algorithms have been examined: the method of steepest descent and the Davidon-Fletcher-Powell method. An example of steepest descent is given below. Basically, the method computes the gradient of the cost function, \( C \), with respect to the unknown parameters and then searches along the negative gradient direction for values of the parameters which reduce the cost.

In Figure 8, the point \( L_1 \) represents the set of parameters which corresponds to the starting trajectory. The arrow points in the direction of the negative gradient of \( C \) (i.e., the direction of decreasing NII). Searching along this direction will yield a new point \( L_2 \) which corresponds to a new trajectory with lower NII. The process of computing gradients and searching continues until the cost converges to within a specified tolerance. In this example, the sequence begins at \( L_1 \) and converges to \( L^* \), where the NII is an absolute (or global) minimum.

Consider, however, the case where the starting trajectory is characterized by the point \( Q_1 \). The optimization process will converge to the point \( Q^* \), which is a relative (or local) minimum. The trajectory characterized by \( Q^* \) does give a lower NII than the starting path at \( Q_1 \), but the NII at \( Q^* \) is still higher than that at \( L^* \).

In this example, it is easily seen that if the starting point lies in Region I, convergence to the global minimum at \( L^* \) is assured (likewise for Region II and the convergence to the local minimum at \( Q^* \)). The
only way to insure that the point \( L^* \) is found is to execute the optimization algorithm a number of times with different starting points, as indicated by the open circles in Figure 8. There is the possibility that the cost function has a "sharp" global minimum, such as at the point \( R^* \). In such a case, it is likely that none of the starting points chosen would result in convergence to \( R^* \). From a practical point of view, though, it is not important that the true global minimum at \( R^* \) is not found. The range of parameters defining the sharp "well" at \( R^* \) is so narrow that a pilot could not deviate from the optimal path characterized by \( R^* \) without greatly increasing the NII. Simply stated, the only optimal path of interest is one whose resulting NII is not overly sensitive to slight variations in the path.

The steepest descent algorithm has the disadvantage of giving slow convergence near the optimal set of unknown parameters; however, significant reduction in the cost (NII) does occur during the first few iterations. A superior algorithm is the Davidon-Fletcher-Powell method, which gives good convergence near the optimum. This method has been employed in this study with satisfactory results. A detailed description of both optimization methods appears in reference 7.

A. The Optimization Algorithm

A computer code has been developed which implements either of the optimization methods described above. Figure 9 shows a flow chart for this code. Initial data (population map, aircraft constraints, initial and final aircraft positions, etc.) are required for each configuration of trajectories and aircraft at a given airport. An initial set of trajectories is either supplied by the user, or a default set is gen-
Figure 9. Flow Chart of the Flight Path Optimization Algorithm
erated by the program. The optimization then begins, with successive values of the cost being compared after each iteration. When the difference between successive values is less than a defined stopping criterion, the process terminates.

The code has been written in modular form so that any of the various models (population distribution, cost function, etc.) may be upgraded or modified easily without making major changes in the code. As an example, the noise impact in each population section requires the computation of an integral. While this integral is usually approximated, a more accurate calculation can be made with the simple addition of a subroutine to the program.

Appendix A contains the FORTRAN code as written for a CDC Cyber 172 machine.

B. Results

All of the cases discussed here involve the Patrick Henry International Airport in Hampton, Virginia. The SITE II program was used to generate the population data for each block as shown in Figure 10. The three entry points referred to, Swing, Franklin, and Cape Charles, are the check points indicated on the ILS approach plate (figure 11).

Reduction in the NII at Patrick Henry Airport is limited by the population distribution. As indicated in Figure 12, most of the people are located in blocks near the runway. During takeoffs and landings, these people will be affected by aircraft noise regardless of the trajectories flown.

(1) The Swing and Franklin entry points are used simultaneously for approach/landings. With the Fourier series model of the flight path,
Population Grid at Patrick-Henry Airport (Partial)

Figure 10. Population Grid Scheme at Patrick-Henry Airport
Figure 11. ILS Landing Approach at Patrick Henry Airport
Figure 12. Fourier Series Representation
Table I  Sine Harmonic Representation of Flight Paths

(I) Sine harmonic series representation of flight path

Two landing trajectories with entry points at Swing and Franklin, respectively. Aircraft distributions on both of the trajectories are:

<table>
<thead>
<tr>
<th></th>
<th>B707</th>
<th>B727</th>
</tr>
</thead>
<tbody>
<tr>
<td>day time</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>night time</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sine harmonic parameter α's</th>
<th>Swing entry trajectory</th>
<th>Franklin entry trajectory</th>
<th>Annoyance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.5074 x 10^2</td>
<td>- 2.8931 x 10^2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-7.5166 x 10^2</td>
<td>5.5448 x 10^2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1.4734 x 10^3</td>
<td>- 7.7248 x 10^2</td>
<td>1.5142</td>
</tr>
<tr>
<td>4</td>
<td>- 1.1312 x 10^3</td>
<td>1.0156 x 10^3</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.2254 x 10^3</td>
<td>- 1.0825 x 10^3</td>
<td></td>
</tr>
</tbody>
</table>

Figure 12
the results obtained are shown in Figure 12. Details appear in Table I. As is easily seen, there is an unnecessary amount of waviness in the trajectories far from the runway. This is caused by the fact that the Fourier series is truncated after five terms. More terms could be included but more computation time would be required. Thus, the line segment model of the flight path has been adopted and is used in all the following cases.

(2) A single trajectory, with one Boeing 707 flying, is determined using the line segment model. The results are shown in Figures 13a and b. Both the Swing and Franklin stations have been used as entry points. There are three segments in each trajectory, requiring only three turning maneuvers from the pilot. This is clearly more realistic than the type of path produced by the Fourier series model. A comparison of the results of the two models shows that the line segment scheme yields slightly higher NII values (3-5% higher than in the Fourier series representation); however, the NII is reduced, compared to existing approach paths, by 4-6%.

(3) Multiple aircraft on multiple trajectories are investigated. Figures 14, 15 and 16 show the results for two, three, and four segments per trajectory. The reduction in NII ranges from 4 to 5%. Details appear in Table II.

(4) The multiple aircraft, multiple trajectory case is repeated (with three segments per trajectory) using Gaussian quadrature to evaluate the integral in the NII computation; a 6% reduction is seen. Figure 17 shows the difference that results when this integration is
Figure 13b Single Aircraft (Landing)
Figure 14a Two Segments per Trajectory (Landing)
Figure 14b. Existing Landing Approach
Figure 15. Three Segments per Trajectory (Landing)
Figure 16. Four Segments per Trajectory (Landing)
Table II  Line Segment Representation of Flight Path

(II) Line segment representation of flight path
(A) Balanced distribution of aircraft on each trajectory*

<table>
<thead>
<tr>
<th>Number of entry points</th>
<th>Number of segments on each trajectory</th>
<th>Corner Points (x,y)</th>
<th>Remark</th>
<th>Figure Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Swing entry trajectory</td>
<td>Franklin entry trajectory</td>
<td>Cape Charles entry trajectory</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>(-25600, 9600)</td>
<td>(-17209, 5797)</td>
<td>(0,0)</td>
</tr>
<tr>
<td></td>
<td>0.8395</td>
<td>(-16115, 0)</td>
<td>(-16115, 0)</td>
<td>(-800, 0)</td>
</tr>
<tr>
<td>2</td>
<td>1.458</td>
<td>(-17209, 5797)</td>
<td>(-16115, 0)</td>
<td>(-800, 0)</td>
</tr>
<tr>
<td></td>
<td>1.401</td>
<td>(-17339, 6371)</td>
<td>(-8785, 5563)</td>
<td>(-800, 0)</td>
</tr>
<tr>
<td>3</td>
<td>1.395</td>
<td>(-17339, 6408)</td>
<td>(-10031, 5752)</td>
<td>(-800, 0)</td>
</tr>
<tr>
<td></td>
<td>1.311</td>
<td>(-17339, 6408)</td>
<td>(-10031, 5752)</td>
<td>(-800, 0)</td>
</tr>
<tr>
<td>4</td>
<td>1.395</td>
<td>(-25600, 9600)</td>
<td>(-19400, 7200)</td>
<td>(-20274, -3917)</td>
</tr>
<tr>
<td>3</td>
<td>1.262</td>
<td>(-25600, 9600)</td>
<td>(-17333, 6400)</td>
<td>(-13181, 4661)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-8911, 3419)</td>
<td>(-6661, 4503)</td>
<td>(-800, 0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-25600, 9600)</td>
<td>(-16115, 0)</td>
<td>(-800, 0)</td>
</tr>
</tbody>
</table>

* Aircraft distribution on each trajectory is:

<table>
<thead>
<tr>
<th>Aircraft</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>B707</td>
<td>2</td>
</tr>
<tr>
<td>B727</td>
<td>2</td>
</tr>
</tbody>
</table>

Day time 2 2
Night time 2 2
Figure 17. Comparison of Trajectories Using Gaussian Quadrature (dotted line) and Centroid Approximation (solid line)
performed as compared to the customary approximation.* Details appear in Table II.

(5) The aircraft mix on the Swing and Franklin trajectories is altered to be unbalanced (details in Table III). Figures 18a and b show that between the two cases of unbalanced aircraft distribution, the resulting change in the trajectories is slight and the change in NII is only 0.07%. This may point to the existence of optimal "corridors" which are independent of the aircraft distribution.

(6) All three entry points are used simultaneously for multiple aircraft, as shown in Figure 19. The important result here is that the optimum trajectory from Cape Charles is found to pass over the water, as should be expected. Details appear in Table II.

(7) Some preliminary work has been done on the takeoff problem. For each takeoff trajectory, the end of the runway becomes the initial point, and the final point (approximately 30 km away) may be placed anywhere. Two final points in the region northwest of the airport and two in the southwest region were chosen. Optimal paths were computed for the different pair combinations (three segments per trajectory). These are shown in Figures 20, 21, and 22 with details given in Table IV. The pair giving the lowest NII is shown in Figure 21. (NII = 1.425).

(8) To help guarantee that the optimal set of trajectories is found by the searching algorithm, a method called "selective search" has been devised. Figure 23 shows a simple version of it. Basically, a number

---

*Referred to as the centroid approximation, since the $L_{dn}$ in a given population block is calculated at the centroid of the block and assumed constant over the entire block.
Table III Unbalanced Distribution of Aircraft

(B) Unbalanced distribution of aircraft on each trajectory

<table>
<thead>
<tr>
<th></th>
<th>Swing entry trajectory</th>
<th>Franklin entry trajectory</th>
<th>Annoyance</th>
<th>Figure Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A/C distribution</td>
<td>Corner Points</td>
<td>A/C distribution</td>
<td>Corner Points</td>
</tr>
<tr>
<td></td>
<td>Day</td>
<td>Night</td>
<td>Day</td>
<td>Night</td>
</tr>
<tr>
<td></td>
<td>B707</td>
<td>B727</td>
<td>B707</td>
<td>B727</td>
</tr>
<tr>
<td>CASE 1</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>(-25600, 9600)</td>
<td></td>
<td>(-26666, -5333)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-17339, 6371)</td>
<td></td>
<td>(-17996, -3549)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-8785, 5563)</td>
<td></td>
<td>(-9150, -3284)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-800, 0)</td>
<td></td>
<td>(-800, 0)</td>
<td></td>
</tr>
<tr>
<td>CASE 2</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>(-25600, 9600)</td>
<td></td>
<td>(-26666, -5333)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-17339, 6371)</td>
<td></td>
<td>(-18025, -3525)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-8785, 5563)</td>
<td></td>
<td>(-9000, -3776)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(-800, 0)</td>
<td></td>
<td>(-800, 0)</td>
<td></td>
</tr>
</tbody>
</table>

*3 line segments on each trajectory
Figure 18a. Unbalanced Aircraft Distribution
Figure 18b. Unbalanced Aircraft Distribution
Figure 19. Three Entry Point Landing Trajectories
Figure 20. Takeoff Trajectories
Figure 21. Takeoff Trajectories
Figure 22. Takeoff Trajectories
### Table IV Takeoff Trajectories

(C) Takeoff Paths*

<table>
<thead>
<tr>
<th>Number of segments on each trajectory</th>
<th>Annoyance NII</th>
<th>Takeoff Trajectories</th>
<th>Figure Number</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exit point 1</td>
<td>Exit Point 2</td>
</tr>
<tr>
<td>3</td>
<td>1.515</td>
<td>(-15000, 25980)</td>
<td>(-15000, -25980)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-13200, 13200)</td>
<td>(-15000, -25980)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-8300, 5000)</td>
<td>(-15000, -25980)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-800, 0)</td>
<td>(-15000, -25980)</td>
</tr>
<tr>
<td>3</td>
<td>1.425</td>
<td>(-15000, 25980)</td>
<td>(-15000, -25980)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-13200, 13200)</td>
<td>(-17655, -10323)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-8200, 5000)</td>
<td>(-17655, -10323)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-800, 0)</td>
<td>(-17655, -10323)</td>
</tr>
<tr>
<td>3</td>
<td>1.455</td>
<td>(-15000, 25980)</td>
<td>(-25980, -15000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-13200, 13200)</td>
<td>(-25980, -15000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-8200, 5000)</td>
<td>(-25980, -15000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-800, 0)</td>
<td>(-25980, -15000)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-8725, -5559)</td>
<td>(-17565, -10030)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-800, 0)</td>
<td>(-17565, -10030)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(-800, 0)</td>
<td>(-17565, -10030)</td>
</tr>
</tbody>
</table>

*Aircraft distribution on each trajectory is

<table>
<thead>
<tr>
<th></th>
<th>B707</th>
<th>B727</th>
</tr>
</thead>
<tbody>
<tr>
<td>day time</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>night time</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>
Figure 23. Selective Search Scheme
of trajectories are evaluated as candidates for the starting set input to the algorithm. The process consists of choosing several segments in different regions of the near-terminal area and evaluating the NII for various combinations of them. The set of trajectories with the lowest NII is taken as the starting set for the optimization routine. This way, it is more likely that the optimal set will be found. Such a selective search corresponds loosely with the choosing of different starting points in the example given earlier (figure 8).

**CONCLUSION**

A method has been formulated which optimizes aircraft paths during approach or takeoff. Multiple aircraft flying on several trajectories can be considered. Models have been developed using available data where possible for the population distribution, aircraft noise signatures, noise impact, constraints, and flight path. An algorithm which uses either the steepest descent method or the Davidon-Fletcher-Powell method for optimization has been implemented and tested. The algorithm can

1) Evaluate the noise impact of existing flight paths,
2) Evaluate the noise impact of proposed flight paths, and
3) Optimize the flight paths to minimize the noise impact, subject to constraints.

The method has been applied to the Patrick Henry International Airport area. Existing paths have been evaluated for noise impact and optimal paths were determined using either two or three of the terminal area entry points. Approximately 4.5% improvement in NII was achieved over that of the paths presently used. The population is concentrated
near the end of the runway, though, and it is felt that even more improvement in the NII could be achieved at other airports.
REFERENCES


PROGRAM MANIP INPUT=TAPE5 OUTPUT=TAPE6 INFUT=TAPE7 TAPE8
PROGRAM PANIP

* CALCULATE INITIAL CORNER POINTS

* APLANE = APLANE+ND727(I)+N707(I)+ND727(I)+N727(I)
  XM(I,J) = X0(I)
  YM(I,J) = Y0(I)
  XM(I,NSEG+1) = XF(I)
  YM(I,NSEG+1) = YF(I)
  XM(I,NSEG+2) = (XPRT+XF(I))/2.
  YM(I,NSEG+2) = (YPRT+YF(I))/2.

IF (CONTINUE.EQ.1) GO TO 20
DO J = 2, NSEG
  XM(I,J) = X0(I)+(FLOAT(J-1)/FLOAT(NSEG))*(XF(I)-X0(I))
  YM(I,J) = Y0(I)+(FLOAT(J-1)/FLOAT(NSEG))*(YF(I)-Y0(I))

CONTINUE

GO TO 20

READ (F09) XM(I,J), YM(I,J), J=2, NSEG)

DO 40 I = 1, NMAP
  DO 40 J = 1, NSEG
    ARRAY(I,J) = 0.

CONTINUE

READ (F09) ARRAY(I,J)

READ (F09) PWET1, PWET2

READ (F09) XMIN, XMAX, YMIN, YMAX

READ (F09) XBEGIN, XFINAL, YDIS, WEIT1, WEIT2

IF (XMIN.LE.ARRAY(I,1).AND.ARRAY(I,1).LE.XMAX. AND. YMIN.LE.ARRAY(I,1))
  CONTINUE
  FLITMAX = APLANE*RATIO

WRITE (6,90) XMIN, XMAX, YMIN, YMAX, ALMAX, FLITMAX

WRITE (6,90) XBEGIN, XFINAL, YDIS

DO 75 I = 1, NTRAJ
  WRITE (6,90) I, XM(I), YM(I), ZM(I)

CONTINUE

WRITE (6,90) ND707(I), N707(I), ND727(I), N727(I)

CONTINUE

NSEG = NSEG + 1

CONTINUE

WRITE (6,90) ND707(I), N707(I), ND727(I), N727(I)
PROGRAM MAIN  73/172  FS

FTH 9.7.669          08/04/24, 10:20:39

115  DO 80  J = 1,NSSEG1
120  CONTINUE
DO 91  I = 1,N

93  CONTINUE

125  CALL GNEWTON (MAXIT,STOPCHG,NSSEG,DELTAX)

130  G

520  FORMAT (2H3X,2HMAXIMUM ITERATION SET TO 12,/,9X,2HNUMBER OF TRAJECTORIES NST=1,/) 20X,4HNUMBER OF POPULATION POINTS ON THE MAP 20X,5HHAIRPOINTER 25X,5HTRAJECTORIES IN Y DIRECTIONS =60X,5H+1.1H METERS FOR 25X,5H+1.1H Meters :/(5X,2HGRADIENT CRITERION FOR GNEWTON, 20X,3H,$/)

420  DO 80  J = 1,NSSEG1

140  40X,3H0.00 FORMAT 420  $12X/5X,24HAREAS WITHIN THE AREA OF X-COORDINATES BETWEEN F6.1,5H AND P6.1,5H TIMES A DAY"></p>
SUBROUTINE COST

SUBROUTINE COST (SGPAO, TOTAL, ANII, PHALTY, CLOSE, THRESH)
COMMON /TRJAJ/ NSAMP, NSEG, XM(1:J), YM(1:J), X1(1:J), Y1(1:J), X2(1:J), Y2(1:J), POSIT(3:100,3), XCENTR(3:100,3), YCLNT(3:100,3), ANGLE(3:100,3), XT(1:J), YT(1:J)
COMMON /THRESH/ ALNAX, LIPTAX, XEIT1, XEIT2, NPLANE
COMMON /LOCAL/ X1(1:3), XM(1:3), YM(1:3), XT(1:3), YT(1:3), ANGLE(1:3), POSIT(3:100,3), XCENTR(3:100,3), YCLNT(3:100,3), ANGLE(3:100,3)
COMMON /REAL/ XH, YH, XM(1:3), YM(1:3), XM1, YM1, XM2, YM2, XM3, YM3, XM4, YM4, XM5, YM5
COMMON /CLOSE/ THRESH = 0.0

C FIRST CALCULATE PARAMETERS AT CORNER POINTS
C
C **************************************************************
C
C DO 50 I = J+1, NTRAJ
DO 50 J = 2, NSEG
T12(J,J) = XM(I) - YM(I)
X12(J,J) = YM(I) - YM(J)
X21(J,J) = YM(I) - YM(J)
Y12(J,J) = XM(I) - YM(I)
Y21(J,J) = XM(I) - XM(J)

DO 40 J = 2, NSEG
A = SRT((XM(I,J-1) - XM(I,J))**2 + (YM(I,J-1) - YM(I,J))**2)
B = SRT((XM(I,J-1) - XM(I,J))**2 + (YM(I,J-1) - YM(I,J))**2)
C = SRT((XM(I,J-1) - XM(I,J))**2 + (YM(I,J-1) - YM(I,J))**2)

40 CONTINUE
50 CONTINUE
C **************************************************************
C DO THE TWO SEGMENTS FORM A STRAIGHT LINE?
C
C IF (ABS(A+B-C)EQ2.0.GT.1) GO TO 20
S2G = SRT((XT2(1,J-1) - XM1(1,J))**2 + (YT2(1,J-1) - YM1(1,J))**2)
SUBROUTINE COST

731172 TS

FTN 4,7+49
80/04/24, 10:20:35

XCL=TRI(J,J-1) = YCENTR(J,J-1)
ALONG1 = 4.
ANGLE1(J-1) = PI
GO TO 30

20 A = SRT((XM(J,J-1)-XM(J,J))**2+(YM(J,J-1)-YM(J,J))**2)
B = SRT((XM(J,J)-XM(J,J))**2+(YM(J,J)-YM(J,J))**2)
C = SRT((XM(J,J)-XM(J,J))**2+(YM(J,J-1)-YM(J,J))**2)

ALONG2 = RADIUS/TAN(BETA/2.)
PNALTY = PNALTY+PMETI*(PMAXI*ALONG2-A1)**2+PMAXI*ALONG2

IF (J *E* 2) PNALTY = PNALTY+PMETI*(AMAX1*ALONG2+ALONG2-A3)**2

1 CONTINUE

GO TO 30

C

CALCULATE TANGENTIAL POINTS, CENTER OF RADIUS R AT J-TH CORNER

XT1(J,J) = XM(J,J)+ALONG2*(XM(J,J-1)-XM(J,J))/A
YT1(J,J) = YM(J,J)+ALONG2*(YM(J,J-1)-YM(J,J))/A

D1 = SOR((XT1(J,J)-Y1(J,J)+XY1(J,J))**2+Y1(J,J-1)-YT1(J,J))**2)
D2 = RADIUS/COS(I/2.)

X = XT1(J,J)-Y1(J,J)/2.
Y = YT1(J,J)+D1*(X-XM(J,J))/D1

YCENTR(J,J-1) = XM(J,J)+D1*(Y-YM(J,J))/D1

ALONG1 = ALONG2

D = SOR((XT1(J,J)-XT2(J,J))**2+(Y1(J,J)-Y1(J,J))**2)

ANG = AACOS((RADIUS-RADIUS)/D)

SEG = SEG+ANG*(J-1)*RADIUS

CONTINUE

A = SOR((XM(J,J)-XM(J,NSEG)-XM(J,NSEG+3))**2+(YM(J,J)-YM(J,NSEG)-YM(J,NSEG+1))**2)

B = SOR((XM(J,NSEG+1)-XM(J,NSEG+3))**2+(YM(J,J)-YM(J,NSEG+1))**2)

C = SOR((XM(J,J)-XM(J,NSEG+1))**2+(YM(J,J)-YM(J,NSEG+1))**2)

ANG = AACOS(D/C)

PNALTY = PNALTY+PMETI*(AMAX1*COS(I/2.)**2)*PMETI

1 CONTINUE

PHALTY = PHALTY+PMETI*(AMAX1*COS(I/2.)**2)*PMETI

1 CONTINUE

CONTINUE

RETURN IF DYNAMIC CONSTRAINT NOT SATISFIED

CONTINUE

IF (PNALTY.E0.) GO TO 60
SUBROUTINE COST 73/172 TS F11 4.7x+95 80/08/24 10:20:35

115 TOTAL = PENALTY
RETURN

120 C : TAKE SAMPLES ALONG THE TRAJECTORY
RETURN

60 DO J=1,MTAJP
DELPJ = E02(IJ)-ZP(J)/NOS(J)
K = 1
POS1(J,K) = XM1(J,K)
POS2(J,K) = YM(J,K)
POS3(J,K+1) = ZI(J)
DO 120 L = 1,NSC(J)
DELX = (XT2(J,K) + XT1(J,L))/NSAMP
DELY = (YT2(J,K) + YT1(J,L))/NSAMP
POS1(J,K+1) = POS1(J,K) + DELX
POS2(J,K+1) = POS2(J,K) + DELY
POS3(J,K+1) = POS3(J,K) + DELPM(J)*SQR(DELX**2 + DELY**2)
K = K+1
70 CONTINUE
IF (ANGLE(J-1).EQ.0) GO TO 100

130 C : SAMPLE THE CORRECT ARC ON THE J-1 TH CIRCLE
1
ALFA1 = ATAN2((YT1(J,J)-YCENTR(J,J)),(XT1(J,J)-XCENTR(J,J))
         )/PO1
AA = ALFA1 + ANGLE(J-1)/4.
XZ = XCENTR(J,J) + RADIUS*COS(AA)
YZ = YCENTR(J,J) + RADIUS*SIGN(AA)
AD1 = SQRT((XT2(J,J)-XZ)**2 + (YT2(J,J)-YZ)**2)
AD2 = SQRT((XT1(J,J)-XZ)**2 + (YT1(J,J)-YZ)**2)
IF (AD1.GT.AD2) IPLUS = -1
II = G+1
IF (II.LE.G) GO TO 90
II = II+1
AA = ALFA1 + ANGLE(J,J)/4.
POS1(J,K+1) = XCENTR(J,J) + RADIUS*COS(-AA)
POS2(J,K+1) = YCENTR(J,J) - RADIUS*SIGN(-AA)
POS3(J,K+1) = POS1(J,K+1) - DELZP(J)*D
K = K+1
GO TO 80
90 POS1(J,K+1) = XT1(J,J)
POS2(J,K+1) = YT1(J,J)
POS3(J,K+1) = POS1(J,K+1) - DELZP(D)
POS1(J,K+3) = POS1(J,K+1) - DELZP(D)
POS2(J,K+3) = POS2(J,K+1) - DELZP(D)
POS3(J,K+3) = POS3(J,K+1) - DELZP(D)
GO TO 60

54
SUBROUTINE COST 73/172 TS

140  K = K+1
150  CONTINUE
170  DELA = (1721+K)SEG-M(1+K)SEG+1)/NAMP
180  DELT = (1721+K)SEG-M(1+K)SEG+1)/NSAMP
200  DO 111 W(K = 1)NSAMP
210  POS1T(J,K+3) = POS1T(J,K+3)-DELT
220  POS1T(J,K+2) = POS1T(J,K+2)-DELT
230  POS1T(J,K+1) = POS1T(J,K+3)-DELT+SORT(DELX*2+DELY*2)
111  K = K+1
120  CONTINUE
130  NPOS1T(J) = K
120  CONTINUE
140  IF (NTRAJ EQ 1) GO TO 270
150  DO LW = 1 NMAP
160  DO 170 J = 4 5
170  DO 180 K = 3 4
180  LOCAL(I,K) = D0
190  ARRAY(I,J) = LOCAL(1,K)
130  CONTINUE
140  DO ZUG = 1 NTRAJ
150  DO 160 K = 4 NMAP
160  LOCAL(K,Z) = D0
170  LOCAL(K,Z) = LOCAL(K,Z)
180  CONTINUE
190  NPOS1T = NPOS1T(J)
200  DO 190 J = 1 NPOS1T
210  IF (ARRAY(K,J) EQ 0) GO TO 190
220  RANGE = SQRT((POS1T(J,K+3)-ARRAY(K,J))**2+(POS1T(J,K+2)-ARRAY(K,J))**2)
230  AL707 = 12.5+10LOG10(3.261*RANGE/500.)
240  AL727 = 11.5+10LOG10(3.281+RANGE/30.)
250  IF (AL707 LE LOCAL(K,Z)) GO TO 150
260  LOCAL(K,Z) = AL707
270  CONTINUE
280  DO 190 K = 4 NMAP
290  LOCAL(K,Z) = LOCAL(K,Z)
300  CONTINUE
310  NLOCAL = NLOCAL(J)
320  DO 220 J = 1 NLOCAL
330  IF (ARRAY(K,Z) EQ 0) GO TO 220
340  RANGE = SQRT((POS1T(J,K+3)-ARRAY(K,Z))**2+(POS1T(J,K+2)-ARRAY(K,Z))**2)
350  AL707 = 12.5+10LOG10(3.261*RANGE/500.)
360  AL727 = 11.5+10LOG10(3.281+RANGE/30.)
370  IF (AL707 LE LOCAL(K,Z)) GO TO 150
380  LOCAL(K,Z) = AL707
390  CONTINUE
400  DO 190 K = 4 NMAP
410  LOCAL(K,Z) = LOCAL(K,Z)
420  CONTINUE
430  IN LOCAL = COL 3+5 = NO. OF OCCURANCES HIGHER THAN ALMAX
440  COL 8 = TOTAL VIOLATING NOISE = ALMAX FOR TRAJ.
450  DUE TO TRAJ. NUMBERS 1-3 RESPECTIVELY
460  END

55
<table>
<thead>
<tr>
<th>Subroutine</th>
<th>73/172</th>
<th>TS</th>
<th>FTH 47489</th>
<th>80/04/24 10:28:35</th>
</tr>
</thead>
</table>

**C**

| 233 | IF (ARRAY(K,3)*EQ.3) GO TO 190 | 4849 |
| 234 | IF (ARRAY(K,6)*EQ.1) GO TO 190 | 4850 |
| 235 | IF (LOCAL(K,1)+2)*EQ.LOCAL(K,1) GO TO 170 | 4860 |
| 236 | LOCAL(K,1+5) = (LOCAL(K,1)+LOCAL(K,1)*10)**(LOCAL(K,1)+LOCAL(K,1)) | 4890 |
| 237 | LOCAL(K,1+2) = (LOCAL(K,1)+LOCAL(K,1))*LOCAL(K,1) | 4910 |
| 238 | LOCAL(K,1)+21 = LOCAL(K,1)+2 | 4930 |
| 239 | LOCAL(K,1+7) = LOCAL(K,1)+LOCAL(K,1)*10 | 4940 |
| 240 | CONTINUE | 4950 |

**C**

| 241 | DO 210 J = INTترا | 4960 |
| 242 | IF (ARRAY(K,1)*EQ.0) GO TO 210 | 4970 |
| 243 | ARRAY(K,1) = ARRAY(K,1)+1*INTترا | 4980 |
| 244 | CONTINUE | 4990 |

**C**

| 245 | CHECK THRESHOLD NOISE CONSTRAINTS | 5000 |
| 246 | DO 210 K = INTترا | 5010 |
| 247 | IF (ARRAY(K,1)*EQ.0) GO TO 210 | 5020 |
| 248 | ARRAY(K,1) = ARRAY(K,1)+1*INTترا | 5030 |
| 249 | CONTINUE | 5040 |

**C**

| 250 | TO COMBINATION EFFECT OF MULTIPLE FLYOVER | 5050 |
| 251 | DO 210 K = INTترا | 5060 |
| 252 | IF (ARRAY(K,1)*EQ.0) GO TO 210 | 5070 |
| 253 | ARRAY(K,1) = ARRAY(K,1)+1*INTترا | 5080 |
| 254 | CONTINUE | 5090 |

**C**

| 255 | CHECK FOR THRESHOLD NOISE VIOLATION AT EACH BLOCK DUE | 5100 |
| 256 | DO 210 K = INTترا | 5110 |
| 257 | IF (ARRAY(K,1)*EQ.0) GO TO 210 | 5120 |
| 258 | ARRAY(K,1) = ARRAY(K,1)+1*INTترا | 5130 |
| 259 | CONTINUE | 5140 |

**C**

| 260 | DO 210 K = INTترا | 5150 |
| 261 | IF (ARRAY(K,1)*EQ.0) GO TO 210 | 5160 |
| 262 | ARRAY(K,1) = ARRAY(K,1)+1*INTترا | 5170 |
| 263 | CONTINUE | 5180 |

**C**

| 264 | TO COMBINATION EFFECT OF MULTIPLE FLYOVER | 5190 |
| 265 | DO 210 K = INTترا | 5200 |
| 266 | IF (ARRAY(K,1)*EQ.0) GO TO 210 | 5210 |
| 267 | ARRAY(K,1) = ARRAY(K,1)+1*INTترا | 5220 |
| 268 | CONTINUE | 5230 |

56
SUBROUTINE COST 73.172 75  FTM 4.7485 80/04/24, 10:23:35

IF (LOCAL(I)+21.GT.FLITMAX) GO TO 260
CONTINUE
SUM = SUM+LOCAL(I)+J)
CONTINUE
THRESH = THRESH+TWEIT1+AMAX1(0.,(EVOLA-FLITMAX)**2)+TWEIT2*AMAX1
1 AMAX(1:*SUM)**2
CONTINUE

C******************************************************************************
C CHECK SEPARATING CONSTRAINTS  
C******************************************************************************
IF (NTRAJ.NE.11) CALL CROSSOR (NPOSIT,NPOSIT,CLOSE)
GO TO 32C
DO 290 J = 1,NMAP
LOCAL(I,J) = 0
CONTINUE
DO 290 J = 1,NMAP
LOCAL(I,J) = LOCAL(I,J)+LOCAL(I,J+1)
CONTINUE
DO 350 I = 1,NMAP
IF (ARRAY(I).EQ.0.) GO TO 300
ARRAY(I) = ARRAY(I)+1
CONTINUE
DO 310 J = 1,NMAP
IF (ARRAY(J).EQ.0.) GO TO 300
ARRAY(J) = ARRAY(J)+1
CONTINUE
DO 350 I = 1,NMAP
IF (ARRAY(I).EQ.0.) GO TO 300
ARRAY(I) = ARRAY(I)+1
CONTINUE

C TOTAL + ANI1=PHALTY+THRESH+CLOSE
IF (IGRAD.LT.0) RETURN
DO 350 I = 1,NMAP
XT(I) = XT(I,1)
CONTINUE
350  CONTINUE
320  CONTINUE
310  CONTINUE
300  CONTINUE
350  CONTINUE
340  CONTINUE
SUBROUTINE COST 721/172 TS

503000

5450

5450 CONTINUE

5450 RETURN

END
SUBROUTINE COST1

CALL COST (IGRAD, F, ANII, PNALTY, CLOSE, THRESH)
RETURN
END

416008 CM STORAGE USED .187 SECONDS
FUNCTION AACOS T9/172 TS

FUNCTION AACOS (A,B,C)
X = (A*B+C+2)/(2.*A*B)
IF (ABS(X).LT.1.) GO TO 10
WRITE (6,4010) A,A,B,C
STOP
10 IF (X.GT.1.) X = 1.
IF (X.LT.-1.) X = -1.
AACOS = ACOS(X)
RETURN

9010 FORMAT (29H TROUBLE IN AACOS, A,A,B,C = Z(E16.9,3X))
END

469WB CM STORAGE USED .995 SECONDS
SUBROUTINE MONIT (IT, X, N, PX, ANII, PNALTY, CLOSE, THRESH)
COMMON /PRINT/ SXCENTR(I), SXCENTR(I), SXT1(I), SYT1(I), SXT2(I), SYT2(I)
DIMENSION X(N)
DO 1 J = 1, NSEG
  L = (I-1)*NSEG + (J-1)*2 + K
  IF (K.EQ.1) XM(I,J) = X(L)
  IF (K.EQ.2) YM(I,J) = Y(L)
  CONTINUE
1 WRITE (0,910) IT, ANII, PNALTY, CLOSE, THRESH
DO 2 J = 1, NTRAJ
  WRITE (6,920) J, XM(I,J), YM(I,J)
  PII = 4*ATAN(1./XJ)
  DO 3 J = 1, NSEG
    WRITE (6,930) XM(I,J), YM(I,J), SXCENTR(I), SYCENTR(I), SXT1(I), SYT1(I), SXT2(I), SYT2(I), SANGLE(I)
  CONTINUE
2 WRITE (6,940) XM(I,NSEG), YM(I,NSEG), NFOSIT(I)
3 WRITE (6,950) (SPOSIT(I,J,K), I = 1, 3, J = 1, NPOSIT(I))
30 CONTINUE
RETURN
C 61 FORMAT (13X,9X,10X,11HITERATIONS, 14X,9X,10HCOSTS, 14X,9X,10HPENALTY ON DYNAMIC CONSTRAINTS, 14X,9X,10HPENALTY ON SEPARATING CONSTRAINTS, 14X,9X,10HPENALTY ON THRESHOLD NOISE, 14X,9X,10HFLIGHT PATH NOISE, 14X,9X,10HANGLE DEGREE, 14X,9X,10HCORNER OF CIRCLE, 14X,9X,10HTANGENTIAL PTS., 14X,9X,10HINTANGENTIAL PTS., 14X,9X,10HCENTRE OF CIRCLE)
C 62 FORMAT (13HPI, 5F8.1, 13HPI, 5F8.1, 13HPI, 5F8.1)
C 63 FORMAT (3(F8.1,13HPI, 5F8.1,13HPI, 5F8.1))
9 END
41cio93 cm STORAGE USED .527 SECONDS
SUBROUTINE RESULT 7/3/72  TS FTN 6.7.685 06/04/72 10:20:39

SUBROUTINE RESULT (ITK, NMAP, NLNS1, PENALTY, CLOSE, THRESH)
COMMON NTRAJ, NMAP, NSEG, XM(3,6), YM(3,6), ARRY(1576,9), SMAPIT(1300, 30840)
DIMENSION XM(3,6), YM(3,6), NPOS1(3)
DIMENSIONxFCENTR(3,3), SYCENTR(3,3), XSYT1(3,3), SYT1(3,3), XSYT2(3,3)
DIMENSION X(NTRAJ, NSEG - 1)
DIM XII = NTRAJ
NSEG = NSEG - 1
DIM XII = NSEG
DIM lO = K = 1
L = 1 + 1 + (I = 1 + (J = 1) + K = K)
IF (K = 2) YM(I,J,K) = X(I,K)
IF (K = 2) YM(I,J,K) = X(I,K)
CONTINUE
WRITE (6,9050) (ARRY(I,J,K) = 1 + K) = 1 + NMAP
RETURN
C FORMAT (1H1X, 10HFLIGHT PATH NO$, 14X, 10HCORNER PT$, 14X, 10HCENT$)
C FORMAT (14X, 1H(PF8.1, PF8.1) 1 + 5X, 1H(PF8.1))
C FORMAT (12X, 1H(PF8.1, PF8.1, PF8.1, PF8.1))
36 FORMAT (Z(IPE10.3, 1X))
41 FORMAT 4CH STORAGE USED 4.643 SECONDS

10 CONTINUE
WRITE (6,9060) (ARRAY(I,J,K) = 1 + K) = 1 + NMAP
RETURN
C FORMAT (1H1X, 10HFLIGHT PATH NO$, 14X, 10HCORNER PT$, 14X, 10HCENT$)
C FORMAT (14X, 1H(PF8.1, PF8.1) 1 + 5X, 1H(PF8.1))
C FORMAT (12X, 1H(PF8.1, PF8.1, PF8.1, PF8.1))
36 FORMAT (Z(IPE10.3, 1X))
41 FORMAT 4CH STORAGE USED 4.643 SECONDS

5 FORMAT 4CH STORAGE USED 4.643 SECONDS
SUBROUTINE CROSSVR (NP0SIT,FX)
COMMON ATTRA,ARRAY,NSEG,RH(3,4),TR(3,6),NARRAY(57,9),SPSIT(3,160)
63

112(3)*2(3)*NPOSIT(3)
53
COMMON /CROSSYitioner/POS1T(101),POS1T(101),POS1T(101),POS1T(101),POS1T(101)
53
YINTP(XV1,XV2,YV1,YV2*XV1*YV2) = (XV2-XV1)XV1*YV2*(XV2-XV1)*YV2
53
17/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)
53
31/2*(YV2-YV1)/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)/(XV2-XV1)
53
FX = FV
53
SAMPLE = (BEGIN-FINAL)/11.
53
CROSS(1,1) = BEGIN-SAMPLE
53
DO 10 J = 1,10
53
CROSS(1,1) = CROSS(1,1)-SAMPLE
53
10 CONTINUE
53
DO 70 I = 1,11
53
DO 80 J = 1,11
53
NPI = NPOSIT(1,1)
53
DO 90 K = 1,NPI
53
SIGN = (POS1T(I,K) = CROSS(I,J))*(POS1T(I,K+1) = CROSS(I,J))
53
IF (SIGN.EQ.0) GO TO 30
53
AK = K
53
GO TO 30
53
IF (SIGN .NE. 0) GO TO 60
53
CROSS(I,K) = POS1T(I,K)
53
GO TO 30
53
60 CONTINUE
53
IF (K .EQ. 1) CROSS(1,1) = YINTP(CROSS(1,1)) = POS1T(I,1),POS1T(I,2)
53
1 IT(I,2,J) = POS1T(I,2,J),POS1T(I,3,J),POS1T(I,4,J),POS1T(I,5,J)
53
IF (K .EQ. 1) CROSS I,J) = YINTP(CROSS(I,J)) = POS1T(I,K),POS1T(I,K+1)
53
2 IT(I,K,2,J) = POS1T(I,K,2,J),POS1T(I,K+1,2,J)
53
IF (K .EQ. KI) CROSS(I,J) = YINTP(CROSS(I,J)) = POS1T(I,K),POS1T(I,K+1)
53
3 POS1T(I,K,1,J) = POS1T(I,K,1,J),POS1T(I,K,2,J),POS1T(I,K,3,J)
53
4 POS1T(I,K,3,J) = POS1T(I,K,3,J),POS1T(I,K,4,J),POS1T(I,K,5,J)
53
5 CONTINUE
53
DO 70 I = 1,11
53
DO 80 J = 1,11
53
FX = FX+CROSS(I,J)*POS1T(I,J)
53
70 CONTINUE
53
SUBROUTINE CROSSVR (NP0SIT,FX)
COMMON ATTRA,ARRAY,NSEG,RH(3,4),TR(3,6),NARRAY(57,9),SPSIT(3,160)
63
SUBROUTINE CAUSOR

72/172  TS

6W  IF (K.EQ.10) GO TO 100

60    DIS2 = CROSS(JK*K1)-CROSS(JK*K1)

60    IF (DI52.EQ.0) GO TO 100

AA = 1

90    IF (K.IA.EQ.1+LT.10) GO TO 110

DIS3 = CROSS(JK*K1+1)-CROSS(JK*K1+1)

65    IF (G152*DI53.EQ.0) GO TO 110

65    IF (ABS(JSON.LT.ABS(DI53)) DIS2 = DI53

AA = IS+1

GO TO 90

100  CONTINUE

GO TO 120

70  110  FR = FX+CHEITZ*DI53**2

120  CONTINUE

130  CONTINUE

RETURN

END

450900 CM STORAGE USED  4.256 SECONDS
SUBROUTINE GNEWTN (MAXIT,STOPCHG,N,NOX,DELTA0)

5 THIS OPTIMIZATION EMPLOYS SELF-SCALING, RESTARTING
   QUASI-NEWTON METHODS.
   REFERENCES: D.C. Luenberger INTRO. TO LINEAR AND NONLINEAR
   PROGRAMMING; P.204.
   MAXIT: MAXIMUM NUMBER OF ITERATIONS ALLOWED
   STOPCHG: STOP IF PERCENTAGE CHANGE IN SUCCESSIVE COSTS IS
   LESS THAN THIS VALUE
   N: DIMENSION OF THE UNKNOWN X
   NOX: PRESENT OR INITIAL VALUE OF UNKNOWN X

10 DIMENSION XNOW(N+1), DELTAX(N)

20 DIMENSION GNDV(30,1), GNEAT(30,1), P(30,1), Q(30,1)

21 DIMENSION 5X(1,1), FF(30,30), GG(30,30), HH(30,30)

22 DIMENSION 5X(30,30)

23 DIMENSION D(30,1)

24 IT = 0

25 GO TO 4

29 DO 1 I = 1

30 XNOW(I+1) = XTEMP(I+1)

31 CONTINUE

35 WRITE (6,440) PENALTY

40 CALL CUR1ST (IT, XNOW, XNEW, PENALTY, CLOSE*THRESH)

45 RETURN

50 WRITE (6,420) PENALTY

60 DO J = 1

65 CONTINUE

70 CALL FNDG (N,FNDW,XNOW,GG,DELTA0)

80 RETURN

45 STEP 1: SET S = IDENTITY MATRIX AND CALCULATE GRADIENT G

60 DO J = 1

65 CONTINUE

70 CALL FNDG (N,FNDW,XNOW,GG,DELTA0)
C  STEP 2: SET D = -SG

CALL MPLY (N=N1*X+GDW*G30.30.1)
GO TO 1 N
O(I,J) = -O(I,J)
CONTINUE

C  STEP 3: MOST IMPORTANT
  LINE SEARCH ALONG D TO FIND A FA THAT SATISFIES
  IF CUST FUNCTION IS ANALYTIC, SLIGHT MODIFICATION IS
  NEEDED (Simplification: Optional)

CALL LSCH (K=N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,CLOSE)

IF XNOW(1) GT XNOW(3) GO TO 1400
IF XNOW(1) GT XNOW(3) GO TO 100

FAS = FSMALL
GO TO 130

IF (FAS .GT. 1E0) GO TO 140
IF (FAS .GT. 1E0) GO TO 10

RETURN

C  RETURN

WRITE (6,9910) PNALTY
RETURN

IF (111,LT,FASIT) GO TO 140
IF (111,LT,FASIT) GO TO 10

WRITE (6,9910) PNALTY
RETURN

C  RETURN

WRITE (6,9910) PNALTY
RETURN
```fortran
110 IF (PNALTY.GT.3) GO TO 200
   CALL MONIT (IT,FANTY,FSMALL,AN1,FNALTY,CLOSE,THRESH)
   GO TO 210
200 WRITE (4,9420) PNALTY
210 CONTINUE

115 IF (FANTY.LT.0.1) GO TO 200

120 CALL MONIT (IT,FANTY,FSMALL,AN1,FNALTY,CLOSE,THRESH)
125 IF (FANTY.LT.0.1) GO TO 220

130 IF ((FLOAT(IT)/FLOAT(NI)) .LT. FLOAT(IT/NI)) GO TO 220
   PNALTY = FSMALL
   GO TO 60

135 C CONTINUE
C
C
140 CALL MPLY (N,N,0.5,0.3,30,30)
   CALL TNSPOS (30,0,20)
   CALL MPLY (N,N,0.5,0.5,30,30)
   CALL MPLY (N,N,0.5,0.5,30,30)
   CALL MPLY (0.5,0.5,30,30)
   DO 230 J = 3, N
      230 CONTINUE
      PNALTY = FSMALL
   GO TO 60

150 DO 240 I = 1, N
      240 CONTINUE
      GMPH4 = GMPH23
      GO TO 60
```

The subroutine is designed to perform a series of mathematical operations, likely related to solving a system of equations using the Newton-Raphson method, as indicated by the subroutine name `NEWTON`. The code appears to handle cases where certain conditions are met, such as checking if `PNALTY` is greater than 3, or if `FANTY` is less than 0.1, and it updates matrix elements based on these conditions. The subroutine also calls other functions such as `MONIT` which might be used to monitor the progress or conditions of the solution. The code ends by calling the function `FORMAT` with a `41` which suggests it is printing output or logging information.
SUBROUTINE PLY (L,P,M,A,B,C,LDEC,MDEC,NDEC)

* CALCULATE MATRIX MULTIPLICATION C = AB

DIMENSION A(LDEC,MDEC), B(MDEC,NDEC), C(LDEC,NDEC)

DO 10 I = 1,L
  DO 10 J = 1,M
    C(I,J) = 0
  10 CONTINUE

RETURN
END

* 140 SECONDS
SUBROUTINE TRANSPOS (M,N,A,B)

DIMENSION A(M,N), B(M,N)

DO 10 I = 1,M
    DO 10 J = 1,N
    B(I,J) = A(J,I)
10 CONTINUE

END

416008 CM STORAGE USED   .090 SECONDS
SUBROUTINE FGRAD 73/172 TS

SUBROUTINE FGRAD (N,F,G,DELTAX)

C * CALCULATE GRADIENT OF COST F WITH RESPECT TO UNKNOWN X

C

DIMENSIONS X(N+1), G(3G+1), DELTAX(30)

10  DO 11 I = 1,N
   11  X(I+1) = X(I)+DELTAX(I)

CALL COST1 (L+1,F,X,AMSL,PENALTY,CLOSE,THRESH)

15  CONTINUE

RETURN
END

41600A CM STORAGE USED .137 SECONDS
SUBROUTINE GAFA (N,F,DFA,G,X,D)

9

CALCULATE GRADIENT OF F WITH RESPECT TO DFA

DIMENSION XIN(1), D(30,1), XTEMP(30,1)

DO 14 I = 1,N

XTEMP(I,1) = X(I,1)+DFA*D(I,1)

10 CONTINUE

CALL COST1 (1,N,F,F,DTEMP,1,XN,1,PNALTY,CLOSE,FX)

G = (F-F)/DFA

RETURN

END

410408 CM STORAGE USED .104 SECONDS
SUBROUTINE ERROR 73/172 75  

SUBROUTINE ERROR (K)  
WRITE (6,10) K  
STOP  

C  
10 FORMAT (I4,6HAFTER, 11,33H TIMES THROUGH LINE SEARCH, STILL, 37H CA, 10360)  
44ND =1ND AFA WHICH SATISFIES PK>0)  
END  
10380  

41036B CM STORAGE USED  
.036 SECONDS
SUBROUTINE LINESCH (K,M,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY)
DIMENSION X(30,1), D(30,1), XTEMP(30,1), XNOW(30,1)
DIMENSION X(30,1), D(30,1), XTEMP(30,1), XNOW(30,1), AFA(4), PX(13)

C
C CUBIC FIT BY INITIALLY LETTING AFA = 0 AND DAF = .01
C
C
C IF (K,LT,-15) GO TO 60
DMAX = D(1,1)
DO 10 I = 1,N

C
IF (DMAX,LT,ABS(D(1,1))) DMAX = ABS(D(1,1))
10 CONTINUE

C IF (F(1,M),F(2,1)) CALL CHECK (1,N,FNOW,XNOW,D,AFA,DFA,DFA1)

C
C IF (F(1,M),F(2,1)) CALL CHECK (1,N,FNOW,XNOW,D,AFA,DFA,DFA1)

C
C
C CALL GAFA (N,FNOW,D,AFA,DFA,DFA1)

C
C
C AFA2 = AFA1 - IAFAL - AFAO1

C
C
C DAF = 0.01

C
C
C AFAU = AFAI

C
C
C CALL COST (O,N,FSMALL,XTEMP,PNALTY,CLCSE,THRESH)

C
C
C IF (PNALTY,LT,FSMALL) GO TO 50
CALL COST (1,N,FSMALL,XTEMP,PNALTY,CLCSE,THRESH)

C
C
C AFAU = AFAI

C
C
C CALL COST (O,N,FSMALL,XTEMP,PNALTY,CLCSE,THRESH)

C
C
C AFAU = AFAI

C
C
C CALL COST (O,N,FSMALL,XTEMP,PNALTY,CLCSE,THRESH)

C
C
C AFAU = AFAI
SUBROUTINE LINESCH 73/172 TS

115 MIN = 1
170 IF (FMAX > FX(I)) GO TO 160
   FMAX = FX(I)
   MAX = I
160 CONTINUE
   DO 190 I = 1,N
   XCOMP = XNOY(I) + AFA1MINI*D(I)
   IF (ABS(XCOMP-KN(I)) <= 10) GO TO 200
   XTEMP(I) = XCOMP
190 CONTINUE
   CALL COST1 (D,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH)
   CALL COST1 (D,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH)
125 CALL COST1 (D,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH)
   AFA(MAX) = AFA(N)
   GO TO 156
   END

41030B CM STORAGE USED 2.336 SECONDS
SUBROUTINE CHECK 73/172 TS

SUBROUTINE CHECK (ICHECK)N

DIMENSION XNOW(1,1), D(30,1)
WRITE (6,14) FNOW, N(1) XNOW(1,1), D(1,1), I=1,N
IF (ICHECK.EQ.1) STOP
WRITE (6,20) FAFAO, GAFAO, FAFAO
IF (ICHECK.EQ.2) STOP
WRITE (6,30) U1, UU1, U2
STOP

C 10 FORMAT (20X,4H1) THIS IS SUBROUTINE CHECK WHICH GIVES ALL THE
1 FORMATION IN SUBROUTINE LINESCH. // s32x,7HFCN D = 3PE16.9 // s37x
2 HMD = 37X: MD = 125X: 3X: LPE16.9 X: LPE16.9 X: LPE16.9
3 J FORMAT (1//, 20X, THFAO = 3PE16.9, 5X: 8HGAFAO = 3PE16.9, 9X: 8HGAFAO
1.91 U FJRMAI 1//, 5HUU1 = 3PE16.9, 9X: 8HGAFAO = 3PE16.9, 9X: 8HGAFAO
3U FORMAT (1//, 20X, SHU1 = 3PE16.9, 9X: 8HGAFAO = 3PE16.9, 9X: 8HGAFAO
1.9)

419008 CM STORAGE USED .143 SECONDS

ORIGINAL PAGE TO
OF POOR QUA}