(NASA-CR-164719) MINIMUM NOISE IMPACT AIRCRAFT TRAJECTORIES Final Report (Virginia Univ.) 206 p HC A10/MF A01 CSCL 01C
N81-31159
Unclas
G3/03 27333

ABSTRACT

A method has been developed for studying the feasibility of reducing annoyance due to aircraft noise by modifying the flight trajectories over a community. Numerical optimization is used to compute the optimum flight paths, based upon a parametric form that implicitly includes some of the problem restrictions. The other constraints are formulated as penalties in the cost function. Various aircraft on multiple trajectores (landing and takeoff) can be considered. The modular design employed allows for the substitution of alternate models of the population distribution, aircraft noise, flight paths, and annoyance, or for the addition of other features (e.g., fuel consumption) in the cost function. A reduction in the required amount of searching over local minima has been achieved through use of the presence of statistical lateral dispersion in the flight paths.

Three annoyance models have been studied: Noise Impact Index (NII), Level-Weighted Population (LWP), and Highly Annoyed Population Number (HAPN). It is shown that NII is not suitable for comparing different annoyance reduction strategies in any one community. Use of either LWP or HAPN as the criterion indicates that significant reductions in community annoyance are possible through this approach.

The equivalent aircraft noise concept has been developed, by which the time needed for computation of the noise levels is reduced significantly when a large variety of aircraft is under consideration. This concept has also been shown to have applications in the problem of aircraft-trajectory assignments as it relates to noise reduction.

It is found that, generally, the population distribution does not have a dominant influence in determining the optimum set of flight paths when the mixture of aircraft on those paths changes; the optimum set may have to be recomputed for a change in the flight mix.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF SYMBOLS

$\underline{a}$	Arbitrary vector
$A(\underline{x})$	Annoyance function
$\underline{b}$	Linear coefficient (n-vector) in quadratic cost function; arbitrary vector
c	Constant term in quadratic cost function
c_1, c_2	Aircraft noise level coefficients; constants in the aircraft dynamic constraints
$c_{1,eq}, c_{2,eq}$	Equivalent aircraft noise level coefficient.
c_3, c_4, c_5	Constants in the aircraft dynamic constraints
c_{xx}	Aircraft stability coefficients
dB	Decibel
D-F-P	Davidon-Fletcher-Powell (optimization algorithm)
$\underline{d}_k$	Search direction in the k-th iteration
$\hat{\underline{d}}_k$	Unit vector along $\underline{d}_k$
E	Energy of radiated sound; error in the approximate solution $\underline{x}_k$
E_0	Reference sound energy
EPNdB	Effective Perceived Noise decibel
EPNL	Effective Perceived Noise Level
$f(\underline{x})$	Cost function
$\tilde{f}(\underline{x})$	Cost function including penalties
$f_{F,TOF}$	Cost of fuel and time of flight
$\underline{F}(\underline{x})$	Hessian of the cost (n x n matrix)
g	Acceleration due to gravity
$\underline{g}(\underline{x})$	Gradient of the cost function (n-vector); m-vector of constraints

LIST OF SYMBOLS (Continued)

$\underline{h}(\underline{x})$	m-vector of constraints
HAPN	Highly Annoyed Population Number
I	Sound intensity
IAF	Initial Approach Fix
I_0	Sound intensity reference level
IPNL	Integrated Perceived Noise Level
k_0, k_1, k_2, k_3	Constants in the cubic fit line search
l	Length along a flight path
l_1, l_2	Lengths of adjacent trajectory segments
l_k	Length of a rectangular population grid cell
l_T	Total trajectory length
LWP	Level-Weighted Population
m	Mass of an aircraft; number of constraints
$\underline{M}$	Direction modifying matrix (n x n)
n	Number of unknown parameters; number of corner point coordinates; number of population grid cells
N	Number of noise level samples per day
$N_{a/c}$	Equivalent number of aircraft
N_f	Total number of flights per day
NII	Noise Impact Index
NII'	Modified Noise Impact Index
$p(L_{dn})$	Population distribution function
$P_i(\underline{x})$	Penalty function
$P(L_{dn})_i$	Population in grid cell i

LIST OF SYMBOLS (Continued)

PNL	Perceived Noise Level
P_{Total}	Total population included in annoyance calculation
$\underline{g}_k$	n-vector used in modified D-F-P method
q_∞	Dynamic pressure
$\underline{Q}$	Quadratic weighting matrix (n x n)
r	Distance (range) from an aircraft to any point
R	Radius of curvature of a flight path's ground track
S	Wing reference area
$\underline{S}$	Approximation of the inverse Hessian, an (n x n) matrix
V_T	Magnitude of total aircraft velocity
$\bar{V}$	Magnitude of average total aircraft velocity
$W(L_{dn})$	Impact intensity weighting function
$W_{F,TOF}$	Fuel and time of flight weight
W_{HAP}	Weighting function used in HAPN
w_i	Day-night weighting factor
$\underline{x}$	n-vector of unknown parameters
$\underline{x}^*$	Optimum value of $\underline{x}$
x	Corner point, x-coordinate
x_f	Final trajectory point, x-coordinate
x_0	Initial trajectory point, x-coordinate
x_t	Tangent point on trajectory, x-coordinate
$\hat{y}$	Distance between trajectories along an arbitrary direction

LIST OF SYMBOLS (Continued)

y	Corner point, y-coordinate
y_f	Final trajectory point, y-coordinate
y_0	Initial trajectory point, y-coordinate
y_t	Tangent point on trajectory, y-coordinate
z	Corner point, z-coordinate
$\bar{z}$	Average altitude during a turn
z_f	Final trajectory point, z-coordinate
z_0	Initial trajectory point, z-coordinate
z_t	Tangent point on trajectory, z-coordinate
α_k	Search parameter in k-th iteration
α^*	Optimum value of α_k
γ	Flight path angle
γ_c	Climb angle
γ_d	Descent angle
γ_k	Scaling factor in modified D-F-P method
δ	Final opening angle
δa_{max}	Maximum aileron deflection
δr_{max}	Maximum rudder deflection
$\delta r_1, \delta r_2, \delta r_3$	Constants depending upon control surface deflection limits
θ	Angle of straight-line trajectory with respect to fictitious population distribution x-axis
θ_{disp}	Angle of dispersion with respect to mean flight path
$\underline{\mu}$	m-vector of penalty weights

LIST OF SYMBOLS (Continued)

σ_y	Standard deviation in the lateral dispersion of flight paths
σ_z	Standard deviation in the vertical dispersion of flight paths
ϕ_{max}	Maximum roll angle
$\underline{\Phi}(\underline{x})$	m-vector of penalty or barrier functions
ψ	Yaw angle
ψ_{max}	Maximum allowed yaw rate
$\underline{\omega}$	m-vector of penalty or barrier weights

CHAPTER I

INTRODUCTION

The question to be considered is this: Is it possible to achieve a substantial reduction in community annoyance due to aircraft noise through the use of trajectory modifications? While not a complete remedy for the problem of aircraft noise, such modifications will be shown to aid significantly in its solution. What is particularly appealing about this technique of noise control is that its cost is relatively low compared with other methods. This study has produced a system that may be used to analyze the noise effect of aircraft operations on a given population and to minimize that effect (the community annoyance) to the extent possible via modifications in the flight paths.

Historical Background

The problem of severe aircraft noise began in the 1950's with the advent of the commercial jet airliners. Complaints and litigation proceedings followed soon thereafter, with the result that numerous studies were begun. As the use of jet transportation became more extensive, so did the awareness of the effects of its noise upon the environment and people living near airport terminals. Today the aircraft noise issue is perceived by the airline industry as one of its most important problems.

Analyzing the noise effects of aircraft operations is not a wholly objective matter; the physical measurements of sound levels are only part of the analysis. Numerous physical measures of sound noise levels

have been defined, e.g. L_A, L_{dn}.[1] Some of these attempt to include the frequency response of the human ear, while others try to indicate the degree of perceived annoyance. Ref. [1] is a useful compendium of these measures and rating schemes.

Less easy to quantify, and open to many interpretations, are the subjective psychological and sociological effects of the noise, collectively referred to as "noise annoyance." Several attempts have been made to relate the physical measurement to the noise annoyance. Alexander [2] examined some of the relationship models used in studies of the aircraft noise problem. Tarnopolsky [3] has studied the mental health effects from extended exposure to aircraft noise. While Borsky [4] showed that a linear relationship exists between the percentage of people "highly annoyed" and the noise exposure level (L_{dn}) in the range 55 to 80 dB, Kryter [5, p.444] indicates that the relationship is nonlinear (power law) in terms of the maximum EPNL.[2] More recent studies by von Gierke [6] and Goldstein [7] have used a single quantity to represent the average degree of noise annoyance to a community. This measure, the Noise Impact Index (NII) has the properties of intensivity (includes the intensity of the annoyance) and extensivity (includes the

---

[1] Both of these are logarithmic measures of sound power intensity. L_a is the A-weighted sound level, with the weighting based upon the frequency response of the human ear. L_{dn} is the average day-night sound level, based upon the average power density (A-weighted) during a 24-hour period. Nighttime sound levels are weighted more heavily in the average. Chapter III elaborates on these measures.

[2] The Effective Perceived Noise Level (EPNL) is a measure of the sound power density averaged over a short period of time (8 seconds), with spectral weighting. See Appendix A.

extent of the annoyance, i.e. the numbers of people affected). The Noise Impact Index is defined as the weighted sum of all the people affected, divided by the total affected population. Schultz [8] has shown that the proper weighting to be used for the NII is a nonlinear function of the noise exposure level (in L_{dn}).

Methods for reducing the annoyance due to aircraft operations generally lie in one of three categories: 1) noise source modifications, 2) land use control, and 3) aircraft operational control. Noise source modification is more readily used in the design stage of future aircraft than it is in existing ones, where "retrofitting" the engine nacelles with sound absorption material is quite costly. For example, the estimated cost of retrofitting a Boeing 737 with sound absorption material is $202,000, while the cost of replacing the engines' front fans with quieter ones is $1,000,000 [9]. Land use control involves areas experiencing the most noise annoyance near an airport. Such land areas may be purchased by the airport, or perhaps zoned as industrial areas, thus reducing the numbers of people affected. This remedy is not suitable for metropolitan areas where land for housing is in short supply.

The third method for reducing noise annoyance, aircraft operational control, requires a method of predicting noise levels that will be experienced as a result of whatever operational changes are made. Considerable effort has been made in this regard. Padula and Liu [10] examined a mathematical model which treated an aircraft as a moving prolate spheroid that scattered acoustic waves from a trailing point source. Barkhana and Cook [11] presented a model for determining the region on the ground that would experience noise above 70 EPNdB.

Dunn and Peart [12] studied five basic aircraft types: turbojet, turbofan, turboprop, V/STOL, and the helicopter, developing models of both the noise sources and the noise levels contours. These have been incorporated into computer programs for noise level estimation by Crowley et al. [13]. The statistical lateral dispersion of flight paths about the nominally specified trajectories has been analyzed by Filotas [14], who finds that this has a significant effect on noise levels on the ground fairly near to the mean flight paths. The most sophisticated model devised for predicting sound levels is the FAA Integrated Noise Model. A variety of noise ratings may be estimated based upon actual atmospheric conditions and the mixture of aircraft landing and taking-off at a particular site.

The more common methods used to reduce aircraft noise effects through operational controls are the procedures of two-segment approach and thrust cut-back on takeoff. Zalovcik [15] empirically determined that a combination of the two -- a 6 degree slide slope (angle of the flight path with respect to the ground) followed by the standard 3 degree slope, and an associated thrust reduction -- could yield a decrease in sound pressure levels (SPL) of up to 14 dB. Glick et al. [16], simulating aircraft operations near the San Jose Municipal Airport, studied similar operational modifications, including alternate routes for approaches and landings; however, no systematic method for varying the flight paths was employed.

Jacob [17] presented an analytic solution to the problem of minimizing the average effective perceived noise level for landing aircraft confined to a vertical plane. Barkhana et al. [18] showed the

feasibility of computing trajectories in three dimensions, with noise impact as part of the performance measure being minimized. Their method, a combination of dynamic simulation and the use of steepest descents to determine the aircraft control inputs, required large amounts of computer time. A somewhat different method, parameter optimization, has been shown by Rader and Hull [19] to be relevent to the computation of optimum aircraft trajectories. They employed a gradient method to find the angle-of-attack history for supersonic aircraft that results in the minimum time-to-climb trajectory at full power. A fifth-order Tschebycheff polynomial series was used to approximate the angle-of-attack history, with the polynomial coefficients as the parameters to be determined by the optimization scheme.

Jacobson et al. [20] developed a system for determining optimum aircraft landing trajectories (via parametric optimization) with respect to the NII, for Boeing 707 and 727 aircraft. In the work reported here, the system has been expanded to include both takeoff and landing paths, and a variety of commercial aircraft.

Using this system, the current study has been directed to certain essential problems of noise impact reduction through the use of operational controls. First is the matter of which has the dominant influence in determining the noise impact on a community: the particular assignment of aircraft to the various trajectories, or the population distribution. This has an important bearing on the work of the air traffic controllers, who must make the flight path assignments; recomputing the optimum trajectories for a change in the aircraft mixture is, at present, a lengthy process.

Also examined has been the problem of aircraft takeoff. Because they are less severely constrained, trajectories used in taking off are more easily modified to reduce the noise impact than are landing trajectories. The endpoint of a takeoff path is not restricted to be a specific point in space, and the rate of climb is allowed to vary more than the rate of descent in the landing case.

Another problem inherent to trajectory modications is that of restricting the fuel consumed and/or the time of flight. Although reduced noise effects are desired, it would not be acceptable to achieve them at great expense in terms of fuel or time. A study has been conducted on the effects of including the fuel and time of flight in the performance measure.

Previous work [21] relied on NII as the measure of noise annoyance to a community. While NII acts as a good indication of annoyance per person in comparing communities of different sizes, it does not reliably measure the change in annoyance for modified trajectories in any one community. This has prompted the examination of alternate measures and their effects on the optimum trajectories thus computed.

The parameter optimization approach works well for airports of modest size, where the number of trajectories and hence, the number of parameters needed to specify them, is small. For larger airports, the number of parameters required can result in memory requirements that exceed the capabilities of the computer being used. An alternative scheme has been devised for determining the solution in this situation. In addition, the concept of "equivalent aircraft type" has been developed, in which the noise characteristics of all the aircraft on a given

trajectory are lumped together into a single "equivalent aircraft." This transformation has yielded a considerable reduction in the time required to calculate the noise levels experienced by the populace, and as a byproduct, some insight into the sensitivity of these levels to changes in the numbers of aircraft on the flight paths.

Lastly, the penalty function method of including the various constraints in the problem formulation has been examined to determine the correct weighting for the penalties.

Chapter II is an outline of the problem solution method. Following it are detailed descriptions in Chapters III to VII of the models developed and employed in the optimization software. Chapter VIII contains the results, and Chapter IX, the conclusions and recommendations.

CHAPTER II

THE APPROACH TO THE SOLUTION

To examine the feasibility of reducing noise impact on a community using operational changes in aircraft flight paths, several major components of the system must be considered. These include: specifying aircraft trajectories, incorporating passenger, pilot, aircraft, and regulatory restrictions, and searching for "best" paths that reduce the noise impact. In this chapter, each of these areas is addressed in order to show the rationale behind the approach developed for solving the problem.

Although specific mathematical models and functions have been employed in the development, it is not claimed that they are the sole choices. The modularity of the approach allows for perhaps more accurate replacements to be made in any of the areas subsequently discussed.

Any attempt to reduce noise in a community requires some quantitative evaluation (i.e. model) of the effects of the noise. From the model may be extracted a measure, a numerical function, that quantifies the noise effect.[1] Many such models (and measures) have been established, emphasizing different noise effects. For example, the Speech Interference Level (SIL) attempts to quantify the interference of noise with speech communication, while the Percent Hearing Loss (PHL)

---

[1] It is preferred that the measure be a scalar quantity, a single value, for purposes of ranking different solutions to a particular noise problem. A unique method of ranking different collections of values does not exist.

measure is intended to predict that physiological effect for a given range of average noise levels. For the model of overall annoyance of a community because of noise, there are two major considerations: the intensity of the sound levels present and the extent of those levels in terms of the numbers of people exposed to them. Among the measures that attempt to indicate both the intensity and extensity of the noise is the Noise Impact Index (NII), which has been the principal measure used in this study.

The model of annoyance upon which NII is based treats extensity and intensity in the following ways: For annoyance intensity, it is assumed the psychological response to noise depends upon the average noise levels experienced. Nighttime levels are weighted more heavily than are daytime levels. The intensity of annoyance is represented by a nonlinear weighting function of the average sound level. For annoyance extensity, the NII model attempts to indicate the annoyance of the entire community exposed, by including the number of people exposed at different levels. Intensity and extensity are combined in the NII by summing the number of people exposed to a given noise level, multiplied by the intensity weighting of that level. This sum is then normalized by the total population exposed to annoying levels, giving an indication of the average annoyance per person in the community.

Once the annoyance measure has been chosen, a practical way to evaluate it must be devised. In the use of NII, this entails the use of two mathematical models: a noise level model and a population distribution model. Under ideal conditions, it may be assumed with reasonable accuracy that the noise levels from aircraft far away (several hundred

meters) depend only upon the distance from the aircraft.[2] Since the details of that dependence (and the flight paths) are known for all the aircraft operating in the area, it is possible to compute the average noise level at any point on the ground. The spatial distribution of people around an airport may be approximated by dividing the region into a number of smaller areas (cells). United States Census Data can then be used to obtain the number of people in each cell. For given sets of trajectories and the associated aircraft on them, the average sound level in each cell can be calculated. It is then a straightforward matter to calculate the NII for those given conditions.

With a practical implementation of the annoyance measure established, it is then feasible to begin the search for trajectories that result in the minimum possible value of the measure. Traditionally, aircraft trajectory modification has been performed using the calculus of variations or dynamic programming to solve the equations of motion of the aircraft, subject to certain constraint and minimize some criterion referred to as the cost function. Typically the cost function has a physical significance, such as the total fuel consumed by the aircraft flying these trajectories. While this method of dynamic simulation works well for aircraft confined to a vertical plane (see for example Erzberger [22]), the complexity of the equations in three dimensions results in a large increase in the amount of time needed for computation.

---

[2] This also assumes given power and control settings on the aircraft.

An alternative technique has been applied here. Each trajectory is modelled as a specific functional in parametric form. The optimization process then consists of searching for the set of parameters that specifies the optimum set of trajectories, i.e., the set that results in the minimum value of the cost function. In this transformed version of the problem, the number of variables to be determined, and hence, the computation time, is smaller than in the original form, but the presence of various restrictions on the trajectories now creates some difficulty. Whereas these restrictions may be readily handled in the simulation technique and in fact are often automatically built into the problem formulation, they must be explicitly included when employing the parameter optimization scheme.

Two classes of optimization methods exist for solving this type of problem. The first, referred to as primal methods, searches for the set of parameters that will result in the optimum solution, but the search is conducted over only those sets of parameters that specify acceptable solutions, i.e., no restriction is violated. The second class, approximating methods, searches over any sets of parameters without restriction. Inclusion of the problem constraints is accomplished by adding terms (penalty or barrier functions) to the cost function. The penalty functions are such that the cost function increases when constraints are violated; if no violation occurs, then no penalties are added. The unrestricted optimization will then, ideally, avoid restricted sets of parameters in its search for the lowest value of the cost function. For reasons given in Chapter V, an approximating method was chosen for this work. The unconstrained optimization

method used was the one of Davidon, Fletcher, and Powell, modified by Luenberger [23] to improve its performance in a variety of problems.

As for the formulation of the cost function, clearly the minimization of annoyance is the true objective, given that the presence of the penalties is a necessity; however, other concerns may be included, weighted in proportion to their importance. For example, this work contains a study of the effects upon the solution if the total fuel consumed and time of flight are included.

A suitable parametric form for mathematically describing the trajectories is one that requires a small number of parameters, but that still has sufficient flexibility to satisfy all of the constraints. A good choice for such a trajectory model is a set of linear segments joined by smooth arcs. The types of flight paths that result from this model are similar to those currently flown by commercial airliners. This is desirable from the perspective of the pilots, who must not be burdened with overly complicated flight paths. A trajectory is completely specified once the coordinate of the intersections of its linear segments are determined. This is the result of the optimization process.

Included in the constraints are: the dynamic capabilities of the aircraft, considerations of passenger comfort, maintenance of safe separation distances between two trajectories, avoidance of frequent high noise level flights over any one area, and a limit to the number of maneuvers required of the pilots. If some constraints place different requirements upon some property of the trajectory set, then the most restrictive of the constraints prevails.

In the following chapters, the models and measures used in the solution method are treated in detail. Chapter III develops the annoyance model and considers alternative to NII. Chapter IV derives the trajectory model. The optimization method is described in Chapter V. Finally in this development, the cost function, including the formulation of the constraint penalties, is discussed in Chapter VI.

CHAPTER III

THE ANNOYANCE MODELS

As discussed in the previous chapter, a desirable community noise annoyance model includes both the intensity and extensity of the annoyance and has associated with it a scalar measure that quantifies the annoyance. Three measures are examined in the following sections: Noise Impact Index (NII), Level-Weighted Population (LWP), and Highly Annoyed Population Number (HAPN). First, however, models of aircraft noise level and population distribution, which are required by the annoyance models, are developed.

Aircraft Noise Model

For most types of aircraft, the sound level or intensity (power per unit area) in dB units may be expressed as a function of the distance r from the plane:

$$L = c_1 - c_2 \log_{10} r \tag{3-1}$$

where c_1 and c_2 are constants for a given type of aircraft operating with given power and control settings.

The human ear converts sound energy into nerve impulses with some filtering of certain frequency ranges. In order to account for this, a measure of sound level that includes an approximation of the ear's frequency response has been devised (see Ref. [1]). Known as the A-weighted level, L_A indicates the "instantaneously" perceived[1]

[1] Actually, the time required to perceive a change in sound level is about 0.5 sec. See Kryter [5, p.3].

sound level. There are, however, other measures that incorporate the duration of the noise or the presence of high amplitude pure tones in predicting the judged sound level. Of particular interest here is a scale known as Effective Perceived Noise Level (EPNL), whose units are EPNdB. In developing the noise model for this work, data from the Federal Aviation Administration's Integrated Noise Model (see Ref. [24]) has been employed. These relate the noise level in EPNdB to the distance from a particular type of aircraft. Figure 3.1 shows an example of the data for a DC-8-30 jet approaching for landing on a 3° glide slope. The solid line is the least squares fit of the data. It has the relation

$$\mathrm{EPNdB}_{\mathrm{DC\text{-}8\text{-}30}} = 155.74 - 28.32 \cdot \log_{10} r. \qquad (3\text{-}2)$$

Similar fits were obtained for other types of aircraft to be studied. The results are shown in Table 3.1.

All three of the annoyance indicators to be examined use L_A for calculating the annoyance intensity. The approximate conversion is

$$L_A \approx \mathrm{EPNL} - 13\mathrm{dB}. \qquad (3\text{-}3)$$

This relation is derived in Appendix A, along with the procedure for calculation noise levels when multiple sources are present.

The Population Model

In order to assess the extensity of noise annoyance, a model of the population distribution is required. Some annoyance measures are cast in terms of population distribution as a function of noise level; however, for computational purposes, it is better to start with the spatial distribution of people. This is compatible with the noise model

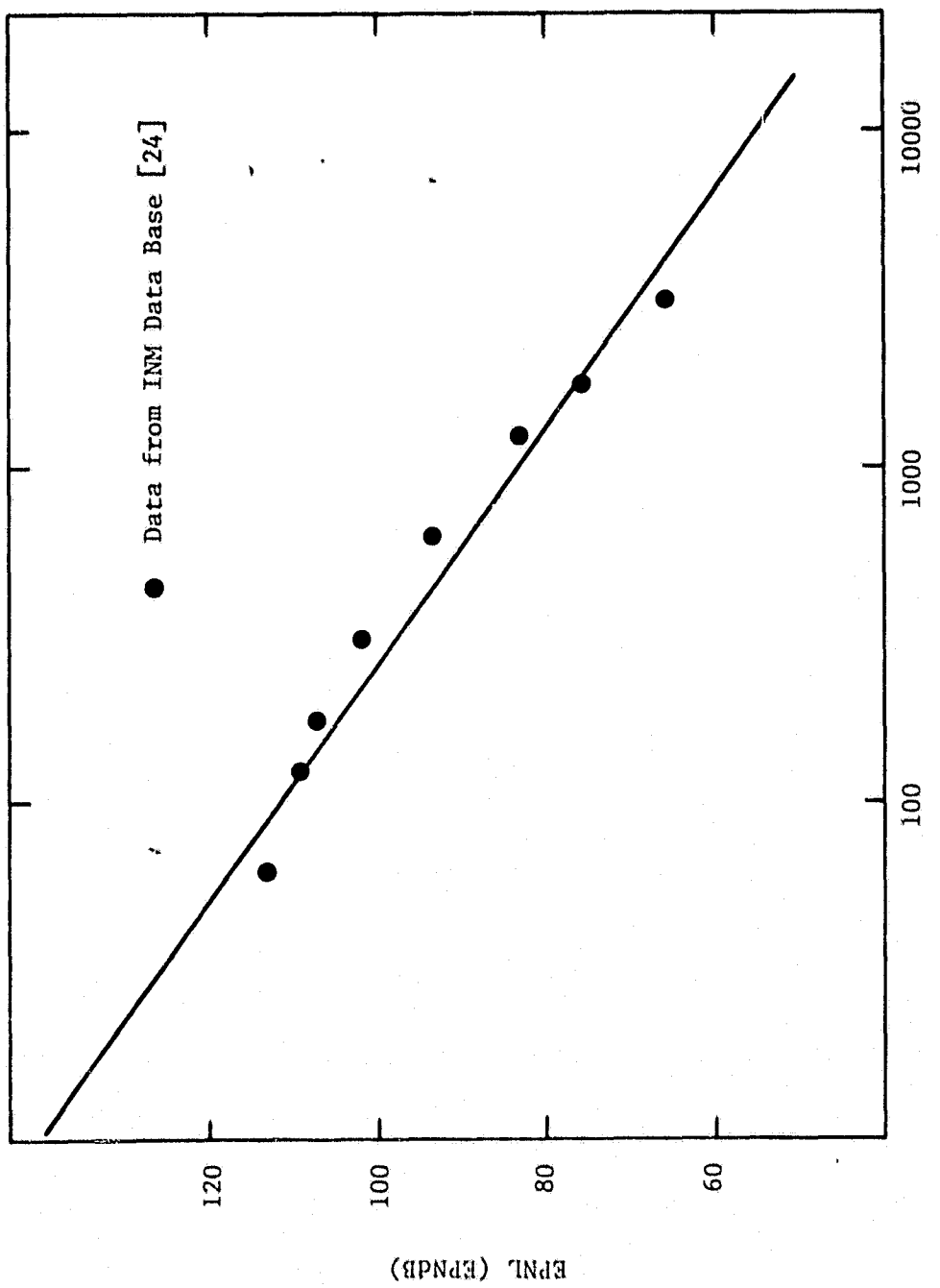


Figure 3.1 EPNL vs. Slant Range for DC-8-30,3° Approach

Table 3.1

L_A vs. Slant Range for Some Commercial Aircraft†

$L_A = c_1 - c_2 \cdot \log_{10} r$, r = slant range (meters)

3° Glide slope appraches

A/C Name	c_1	c_2
DC-8-30	155.74	28.32
DC-9 w/SAM	142.28	24.65
DC-10-10	146.22	28.80
707 w/SAM	135.11	23.60
720	140.88	21.95
727-200	139.47	22.66
727 w/SAM	124.28	18.74
737-100/200	154.74	29.52
737 w/SAM	147.31	26.96
747-200	139.89	24.78
L-1011	136.94	24.49
A-300	166.47	37.20
BAC-111	150.72	27.06
VC-10	142.65	22.95
CV-990	157.34	27.16
SST	136.22	15.96

†Data obtained from Ref. [24] and reduced by least square-error fits.

already developed (the two are appropriately combined later in this chapter) and has the advantage of being independently calculable (i.e. no noise calculations are required).

As described in Chapter II, the distribution is approximated by determining the number of people inside each cell of a grid that overlays a map of the community. The grid should contain a sufficient number of cells to insure good resolution of the distribution, but not so many cells that large numbers of calculations occur in the annoyance measure evaluation. A scheme that satisfies both of these conditions is one in which the size of the cells grows with distance from the airport. The small cells near the airport provide better resolution in a region where the noise levels are changing more rapidly over the ground.

As an example, Figure 3.2 shows the grid used for the Patrick Henry Airport at Hampton, Virginia. (Some of the grid lines in the center are not shown, so that the runways will be visible.) Here, the cells adjacent to the runways are 400 meters on a side. The length of a cell side grows both horizontally and vertically, according to the relation

$$l_{k+1} = 1.3 \cdot l_k \qquad (3\text{-}4)$$

where l_k is the length of cell k in either direction. Covering a square region 32,000 meters on a side, the grid contains 576 cells, approximately 11 times fewer than if all the cells were 400 meters on a side.

Alternative grid schemes also may be used. Some examples are: concentric rings divided into sectors by radial lines (with the ring radii either constant, or growing with distance), and square cells of constant dimensions (if the number required is acceptable).

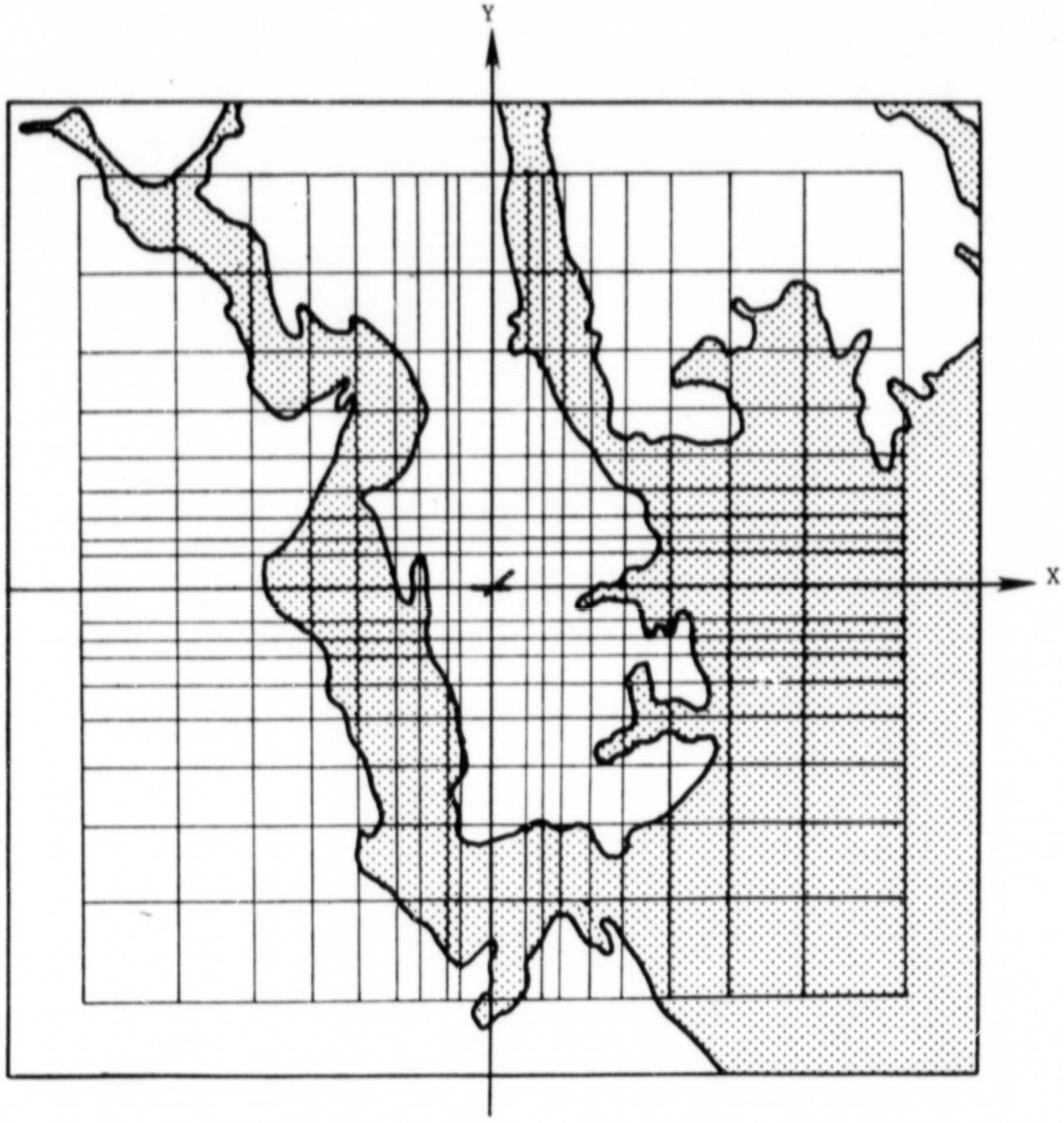


Figure 3.2 Population Grid at Patrick Henry Airport (Partial)

Once the grid scheme is chosen, the number of people in each grid cell may be determined using United States Census Data in a computer program called SITE II [25]. It requires as input a reference point in latitude and longitude, and coordinates with respect to the reference that specify the boundaries of each cell. The population in each cell is output, along with a variety of demographic information. Although only the population data have been used in this study, the other data could be useful in an annoyance measure that includes social and economic information.

In each grid call, it is presumed that the population is concentrated at the geometric centroid of the cell. This is done to simplify the noise level calculations, i.e., all the people will receive the noise level present at the centroid. Chang [26, p.29] has shown that this is a good approximation in that the aircraft trajectories determined using this method are not significantly different from those calculated using more exact methods.

Level-Weighted Population (LWP)

This indicator of community annoyance, devised by the National Academy of Sciences Committee on Hearing, Bioacoustics and Biomechanics [27] assumes that the intensity of annoyance due to noise varies with the noise level. The relation between annoyance and noise level is characterized by the intensity weighting function $W(L_{dn})$, where L_{dn} is

the average day-night noise level. L_{dn} is defined as[2]

$$L_{dn} = 10 \times \log_{10} \frac{\sum_{t=1}^{N} w_t 10^{L_{A,t}/10}}{N} \quad ; \tag{3-5}$$

$$w_t = \begin{cases} 1 \text{ for noise between 7 A.M. and 10 P.M.} \\ 10 \text{ for noise between 10 P.M. and 7 A.M.,} \end{cases}$$

N = Number of noise level samples taken in 24 hours,

$L_{A,t}$ = t-th sample of the noise level, A-weighted.

A plot of $W(L_{dn})$ is shown in Figure 3.3. The analytic expression for it is:

$$W(L_{dn}) = \frac{3.36 \times 10^{-6} \cdot 10^{0.103 \cdot L_{dn}}}{0.2 \cdot 10^{0.03 L_{dn}} + 1.43 \times 10^{-4} \cdot 10^{0.08 L_{dn}}} \tag{3-6}$$

This function is based upon a collection of social surveys of annoyance caused by various types of noise [8]. Although this synthesis respresents possibly the best available data for annoyance prediction, its mathematical model has the property of being unbounded. This is acceptable, though, if there is a bound to the largest value of L_{dn} likely to be encountered.

Level-Weighted Population is defined by

$$LWP = \int W(L_{dn})\, p(L_{dn})\, d(L_{dn}) \tag{3-7}$$

where $p(L_{dn})$ is the number of people receiving noise between L_{dn} and $L_{dn} + d(L_{dn})$. The integration is performed over the range of L_{dn} considered to be annoying. By convention, the lower limit is 55 dB;

---

[2] L_{dn} relies upon the concept of energy addition, which is developed in Appendix A.

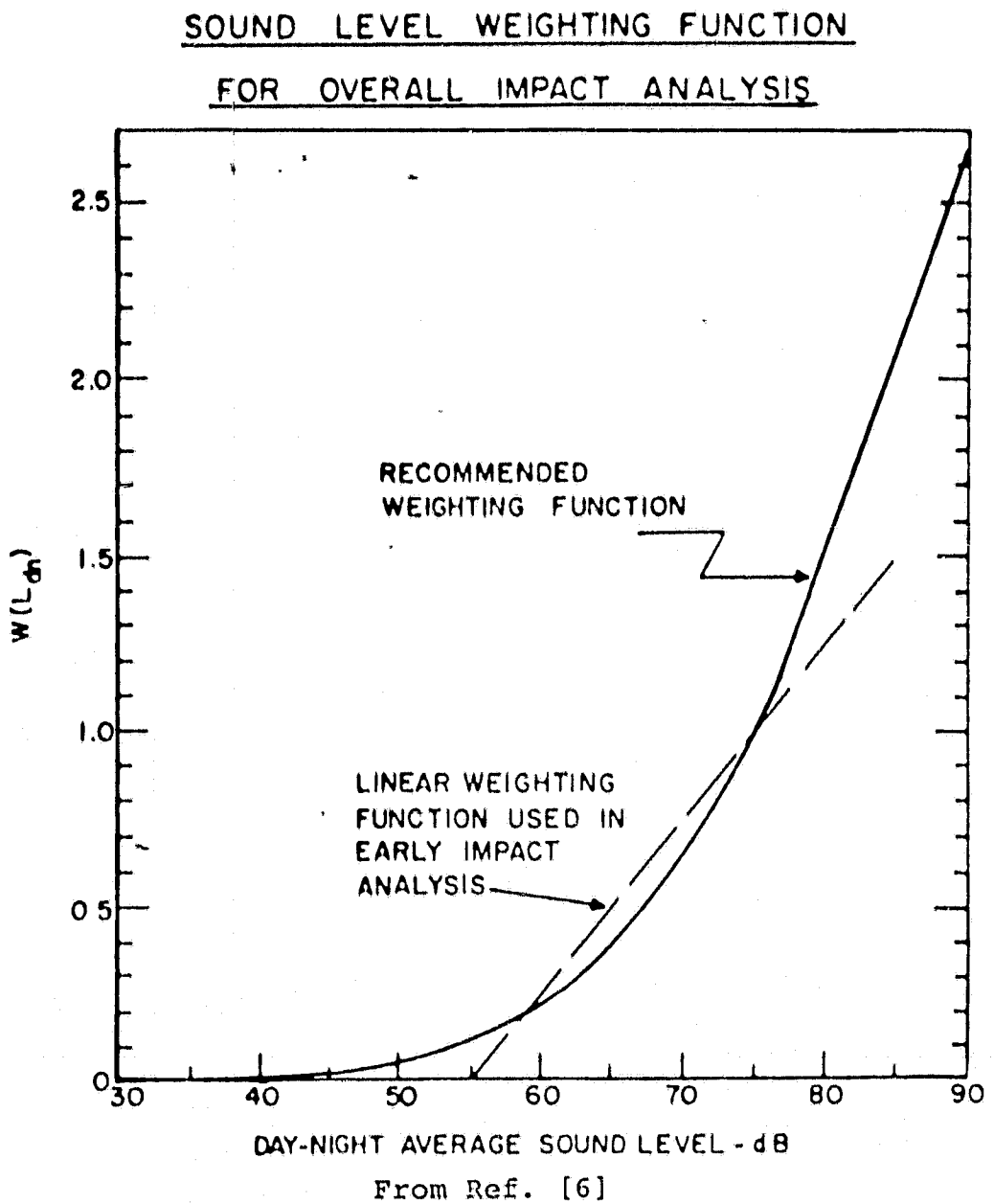


Figure 3.3 Impact Intensity Weighting Function

the upper limit is the highest level present. Evaluation of this integral is facilitated by approximating it as a descrete sum

$$LWP \simeq \sum_{i=1}^{n} W(L_{dn})_i \; P(L_{dn})_i \qquad (3\text{-}8)$$

where i is the index of a population group, $P(L_{dn})_i$ is the number of people in group i, L_{dn} is the noise level they receive and $W(L_{dn})_i$ is the intensity weighting for that L_{dn}.

Noise Impact Index (NII)

This measure, also developed by the National Academy of Sciences, is defined as

$$NII = \frac{LWP}{\int p(L_{dn}) d(L_{dn})} \quad , \qquad (3\text{-}9)$$

where the symbols and limits of integration are the same as in Equation (3-7). Again, a discrete sum is usually used to approximate the integral:

$$\int p(L_{dn}) d(L_{dn}) \simeq \sum P(L_{dn})_i$$

= total population exposed to annoying noise levels.

The purpose of the denominator is to normalize LWP, making NII an indicator of the average annoyance per person in the community. Such an indicator is quite useful when comparing noise problems in communities with widely different populations: (LWP is not a good choice for use in such comparisons because it tends to grow with population size.) There is, however, a serious problem with NII regarding its use in comparing different strategies for noise reduction in any one community. As an example of the problem, consider a simple population distribution of just two clusters of people, as shown in Figure 3.4. The people

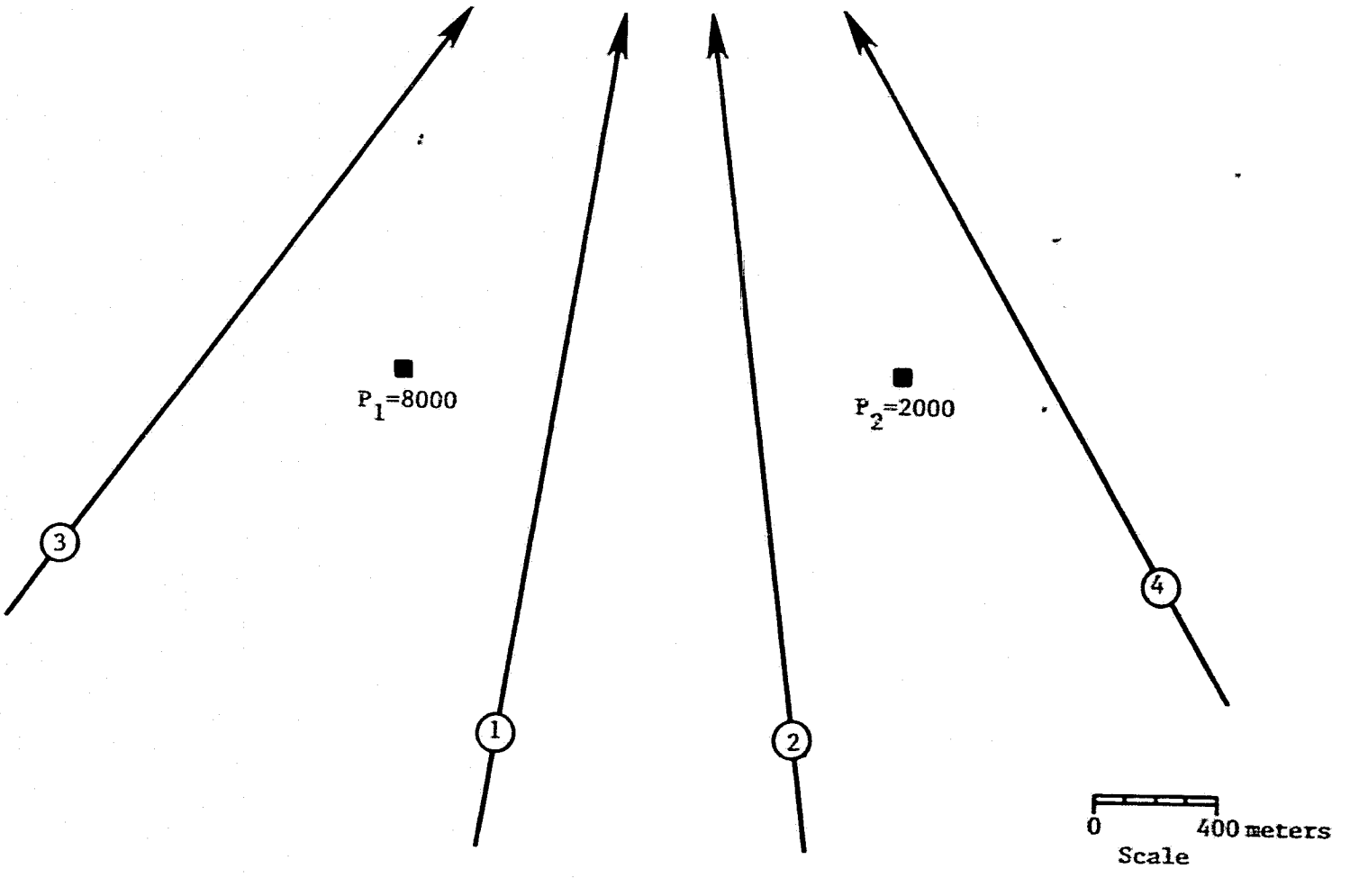


Figure 3.4 A Simple Population Distribution and the Problem With NII

receive noise from aircraft flying on one of the tracks (ground projections) shown. Table 3.2 gives the results for each of the tracks.

Tracks 2, 3, and 4 all have lower LWP than Track 1, but the NII values rank differently. Track 2 has the lowest NII value and Track 4, with the lowest LWP value, has the highest NII. The problem occurs because both LWP and the denominator in Equation (3-9) both change, but by different factors. Clearly, the use of Track 4, where 2,000 people receive an L_{dn} of 67dB is superior (in terms of total community annoyance) to the use of Track 3, where 8,000 people receive an L_{dn} of 64dB. The LWP measure indicates this intuitive choice, while the NII shows the opposite. In Chapter VIII, it will be demonstrated that a similar situation occurs for a real population distribution.

Highly Annoyed Population Number (HAPN)

Perhaps a more tangible indicator of community annoyance would be desirable. An example of such an indicator is HAPN, which attempts to measure the total number of people who are highly annoyed.

The model of annoyance intensity used by HAP differs from that of LWP and NII in two ways: the sound level scale used is maximum dB(A), and the intensity weighting function is:[3]

$$W_{HAP} = \begin{cases} 6.5885 \times 10^{-12} \cdot (L_{Amax})^{5.957603} \\ 6.5885 \times 10^{-12}, \; L_{Amax} \geq 75.3 \text{ dB(A)} \end{cases} \qquad (3\text{-}10)$$

[3] This weighting was obtained by a least square-error fit of data in Kryter [5, p.444].

Table 3.2

Comparison of NII vs. LWP (Simple Population Distribution)

Track	1	2	3	4
	L_{dn}	L_{dn}	L_{dn}	L_{dn}
P_1 = 8000	64	56	64	49.5
P_2 = 2000	56	64	49	67
LWP	3244	1874	2960	1005
NII	0.3244	0.1874	0.3701	0.5026
P_{TOT} †	10,000	10,000	8,000	2,000

†Total population receiving annoying levels (>55 dB)

where

$$L_{Amax} = \begin{cases} \text{maximum A-level received, 7 A.M. to 10 P.M.} \\ \text{maximum A-level received + 10dB, 10 P.M. to 7 A.M.} \end{cases}$$

W_{HAP} is interpreted as the fraction of people receiving the noise who will be highly annoyed.

The definition is

$$HAPN = \frac{\sum_{i=1}^{n} \{W_{HAP}(L_{A,max,day})_i \cdot P_i + W_{HAP}(L_{A,max,night})_i \cdot P_i\}}{P_{Total}} \qquad (3\text{-}11)$$

where the day-night notation is as indicated in Equation (3-5), P_i is the i-th population group, n is the number of groups, and P_{Total} is the total population being considered. Note that daytime and nighttime levels are treated separately, so that it is possible for some population groups to be included twice. Dividing by the total population does not present the same problem here as does the denominator of Equation (3-5), since P_{Total} is a constant for a given community.

CHAPTER IV

THE TRAJECTORY MODEL

Having established in Chapter II that each trajectory should be determined by specifying the unknowns in a general parametric expression, the next matter is to choose that expression, using a form that is compatible with the restrictions involved in the problem. In developing a suitable expression, or model, it is helpful first to examine the sorts of flight paths that are currently used in the region near an airport terminal (within 30,000 meters, where the noise levels are significant). Next, all of the restrictions that pertain directly to the trajectories must be inspected and cast in mathematical terms. Finally, a choice of the parametric form of the flight paths may be made, and the restrictions restated in terms specific to that form.

Typical Flight Paths

A. Landing

In general, an airport will have several runways and be receiving flights from a number of directions. For reasons of safety and organization (from the aspect of air traffic controllers), the number of paths available into the runways is limited by requiring that each aircraft first fly to a reporting point (also referred to as an arrival fix or an entry point). From there it proceeds to its assigned runway, with the requirement that as it passes over a point called the inner marker, its heading must be toward the runway and remain so until touchdown. The direction of all traffic, taking off and landing, on a

given runway is fixed at any given time, determined chiefly by the wind velocity in the area. Figure 4.1 is a reproduction of an approach "plate" used by pilots with instrument landing systems (ILS) in the aircraft. The projection of the trajectory onto the ground is referred to as the "ground track." This particular plate, for Patrick Henry International Airport at Hampton, Virginia, indicates two initial approach fixes (IAF's), Swing and Franklin, for use with the major runway. The entry points and inner markers conveniently serve as endpoints[1] for the landing trajectories that are to be computed.

B. Takeoff

Takeoff paths are considerably less restricted than are landing paths. An aircraft taking off may execute a turn before crossing over the inner marker. It may then proceed along a path that is not required to pass through a reporting point (or "exit fix"). Figure 4.2 shows a typical takeoff path (ground track) for Lambert-St. Louis International Airport.

Except for their different endpoint conditions, landing and takeoff ground tracks are seen to be rather similar: each consists of a combination of straight segments and smooth turns. As for the vertical projection of each type of path (called the "profile"), landing aircraft usually do not descend with a glide slope of more than three degrees, while those taking off may climb at angles as great as fifteen degrees.

---

[1] A fixed entry point is not required for aircraft with more sophisticated navigation equipment, such as VOR/DME (VHF Omni-Range/Distance Measuring Equipment). The flight path model should therefore allow for moveable entry points.

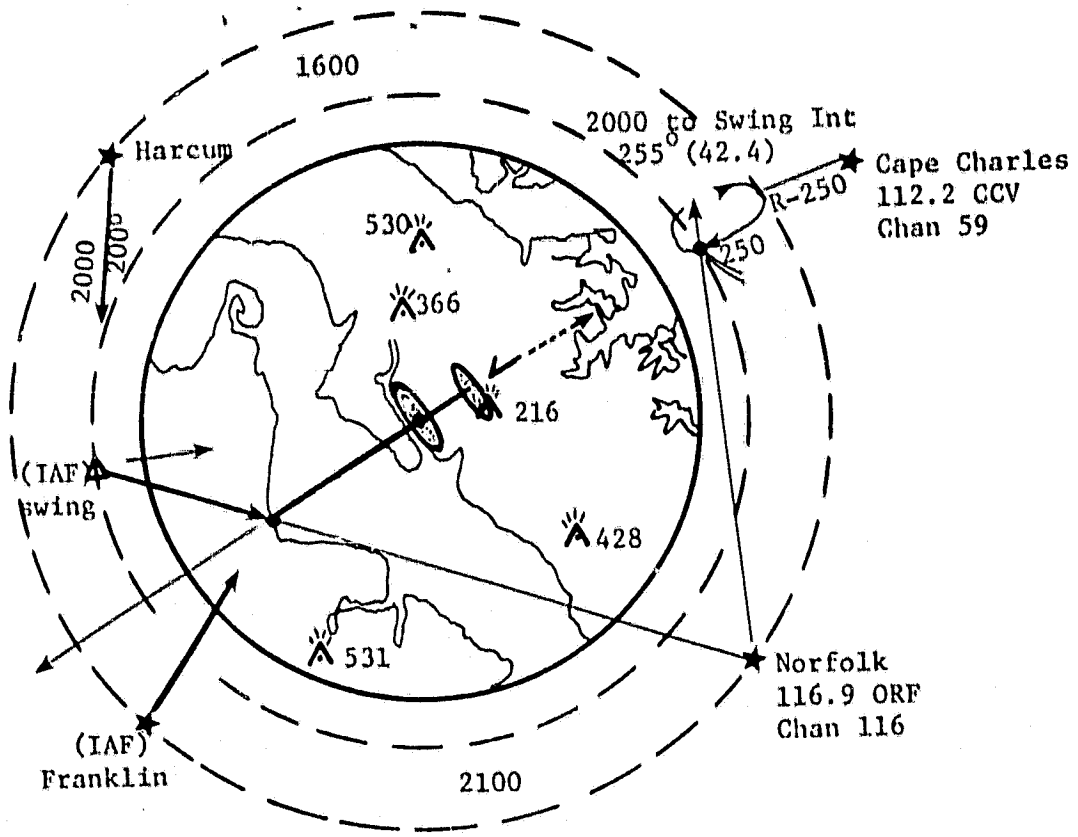


Figure 4.1 ILS Landing Approach at Patrick Henry Airport

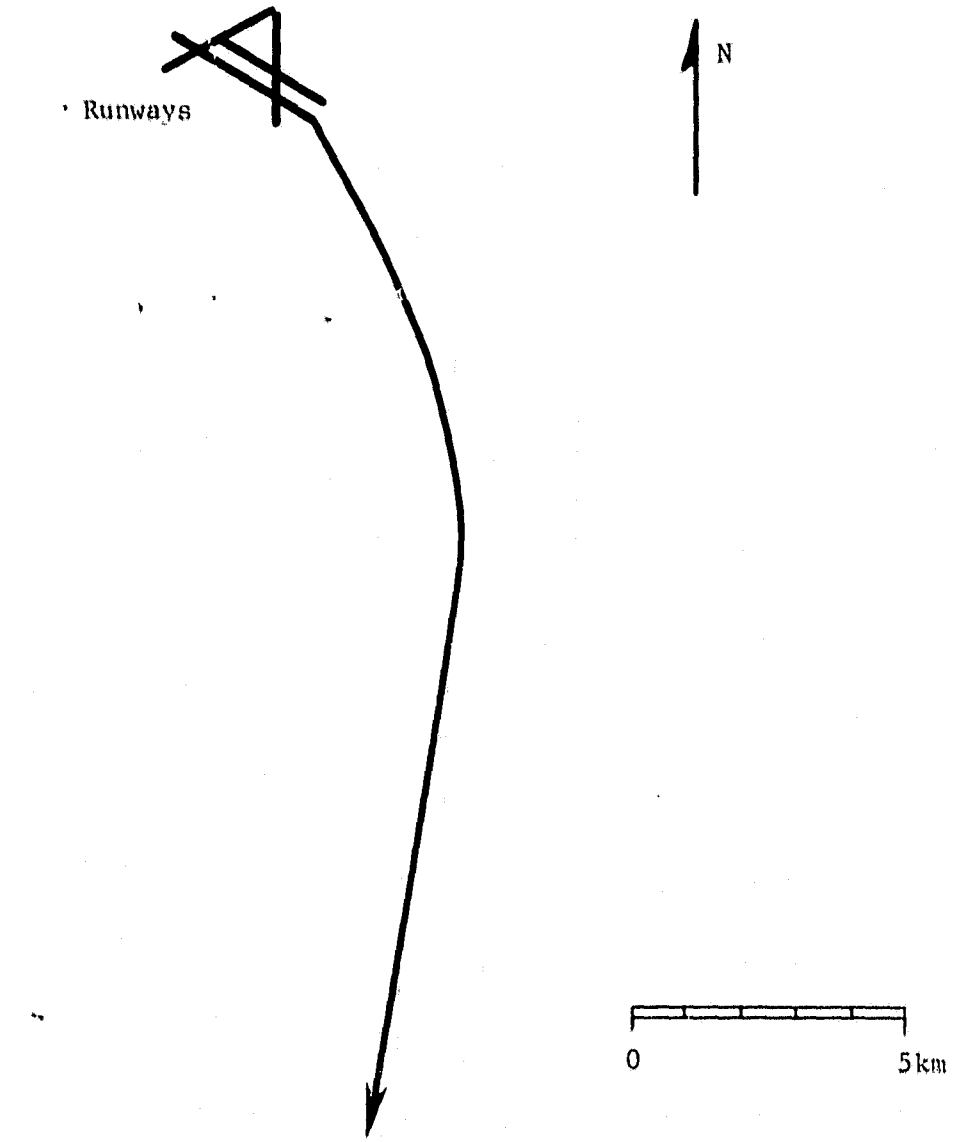


Figure 4.2 Typical Takeoff Trajectory (Ground Track)

Restrictions

The restrictions that pertain directly to the feasibility of the trajectories can be grouped into four categories: aircraft dynamics, passenger comfort, statutory regulations, and piloting considerations. A fifth constraint, concerning the maximum allowable number of high-level noise events for a given population grid cell, has been included in this work.

A. Aircraft Dynamic Constraints

The aircraft dynamic constraints limit a trajectory to one which given types of aircraft are capable of flying. The restrictions may be stated in terms of fixed properties of the aircraft and maximum allowable control surface deflections. In the event of several restrictions on any one quantity, the most severe will prevail. The following expressions, developed from linear equations of aircraft motion [28], are divided into lateral (parallel to the locally flat ground) and longitudinal (in a vertical plane) constraints.

1. Lateral Constraint

$$\dot{\psi}_{max} \leq (c_1 + c_2 c_3) \min(\delta r_1, \delta r_2, \delta r_3) \qquad (4\text{-}1)$$

where

$$\delta r_1 \leq \frac{\phi_{max}}{c_4 + c_2 c_5}$$

$$\delta r_2 \leq \delta r_{max}$$

$$\delta r_3 \leq \frac{\delta a_{max}}{c_2}$$

$$c_1 = - \frac{C_{l_{\delta r}} C_{n_\beta} - C_{n_{\delta r}} C_{l_\beta}}{C_{l_\beta} \bar{C}_{n_r} - C_{n_\beta} \bar{C}_{l_r}}$$

$$c_2 = -\frac{C_{n_{\delta r}} C_{l_r} + C_{l_{\delta r}} C_{n_r}}{C_{n_{\delta a}} C_{l_r} + C_{l_{\delta a}} C_{n_r}}$$

$$c_3 = -\frac{C_{l_{\delta a}} C_{n_\beta} - C_{n_{\delta a}} C_{l_\beta}}{C_{l_\beta} \bar{C}_{n_r} - C_{n_\beta} \bar{C}_{l_r}}$$

$$c_4 = \{C_{y_{\delta r}} (\bar{C}_{l_r} C_{n_\beta} - C_{l_\beta} \bar{C}_{n_r}) + C_{l_{\delta r}} [C_{y_p} \bar{C}_{n_r} + C_{n_\beta} (\bar{m} - \bar{C}_{y_r})]$$

$$+ C_{n_{\delta r}} [C_{l_\beta} (\bar{m} - \bar{C}_{y_r}) + C_{y_\beta} \bar{C}_{l_r}]\} / \frac{mg}{q_\infty S} (C_{l_\beta} \bar{C}_{n_r} - C_{n_\beta} \bar{C}_{l_r})$$

$$c_5 = \{C_{y_{\delta a}} (\bar{C}_{l_r} C_{n_\beta} - C_{l_\beta} \bar{C}_{n_r}) + C_{l_{\delta a}} [C_{y_\beta} \bar{C}_{n_r} + C_{n_\beta} (\bar{m} - \bar{C}_{y_r})]$$

$$+ C_{n_{\delta a}} [C_{l_\beta} (\bar{m} - \bar{C}_{y_r}) + C_{y_\beta} \bar{C}_{l_r}]\} / \frac{mg}{q_\infty S} (C_{l_\beta} \bar{C}_{n_r} - C_{n_\beta} \bar{C}_{l_r})$$

with

$\dot{\psi}_{max}$ = maximum yaw rate

ϕ_{max} = maximum roll angle

δr_{max} = maximum rudder deflection

δa_{max} = maximum aileron deflection

m = mass of aircraft

g = acceleration due to gravity

S = wing reference area

q_∞ = dynamic pressure

C's = aircraft stability coefficients[2]

$\bar{C}$'s = normalized stability coefficients[2]

$\bar{m} = \frac{mV_T}{q_\infty S}$, V_T = total velocity

---

[2] For an explanation of these stability coefficients, see Ref. [29].

The right-hand side of Eqn. (4-1) will be a constant for a given type of aircraft, and the left side may be written in terms of the function that specifies the trajectory. In Figure 4.3a, the ground track is described in Cartesian coordinates as $y = f(x)$. With respect to the arbitrary x-axis, the slope of the flight path is dy/dx, with corresponding angle

$$\Psi = \tan^{-1} dy/dx \tag{4-2}$$

however the quantity of interest is the rate at which that angle is changing (also the yaw rate of the aircraft):

$$\dot{\Psi} = \frac{d}{dt} \tan^{-1} dy/dx = \frac{\partial}{\partial x}(\tan^{-1} \frac{dy}{dx})\frac{dx}{dt} \tag{4-3}$$

But dx/dt is the x-component of the aircraft's velocity (with respect to the ground-fixed coordinate system):

$$V_x = \frac{dx}{dt} = \bar{V} \cos \Psi = \bar{V}\left(\frac{1}{[1 + (dy/dx)^2]^{\frac{1}{2}}}\right) , \tag{4-4}$$

where $\bar{V}$ is the average, total velocity, so that

$$\dot{\Psi} = \frac{(d^2y/dx^2)\bar{V}}{[1 + (dy/dx)^2]^{3/2}} = \frac{\bar{V}}{R(x)} \tag{4-5}$$

where R(x) is the radius of curvature. The constraint (4-1) may now be written

$$\frac{[1 + (dy/dx)^2]^{3/2}}{d^2y/dx^2} = R(x) \geq \frac{\bar{V}}{(c_1 + c_2c_3)\min(\delta r_1, \delta r_2, \delta r_3)} . \tag{4-6}$$

2. Longitudinal Constraint

$$\gamma_{d_{max}} \leq \gamma \leq \gamma_{c_{max}} \tag{4-7}$$

where γ is the angle of the flight path with respect to the horizontal, and $\gamma_{d_{max}}$ and $\gamma_{c_{max}}$ are the maximum allowable angles of descent and

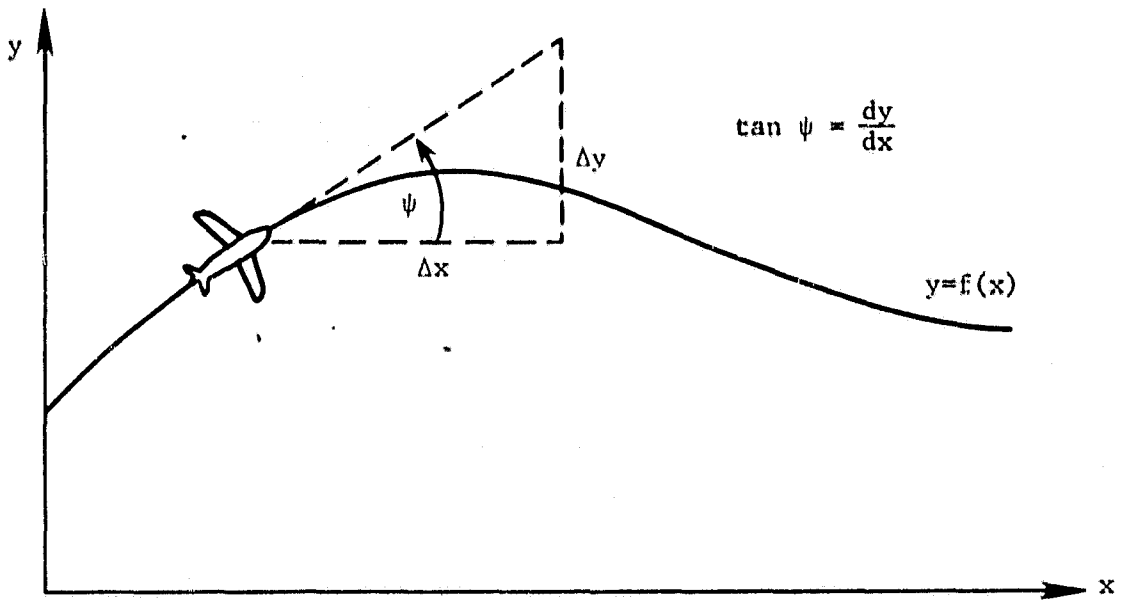


Figure 4.3a Horizontal Flight Path Angle

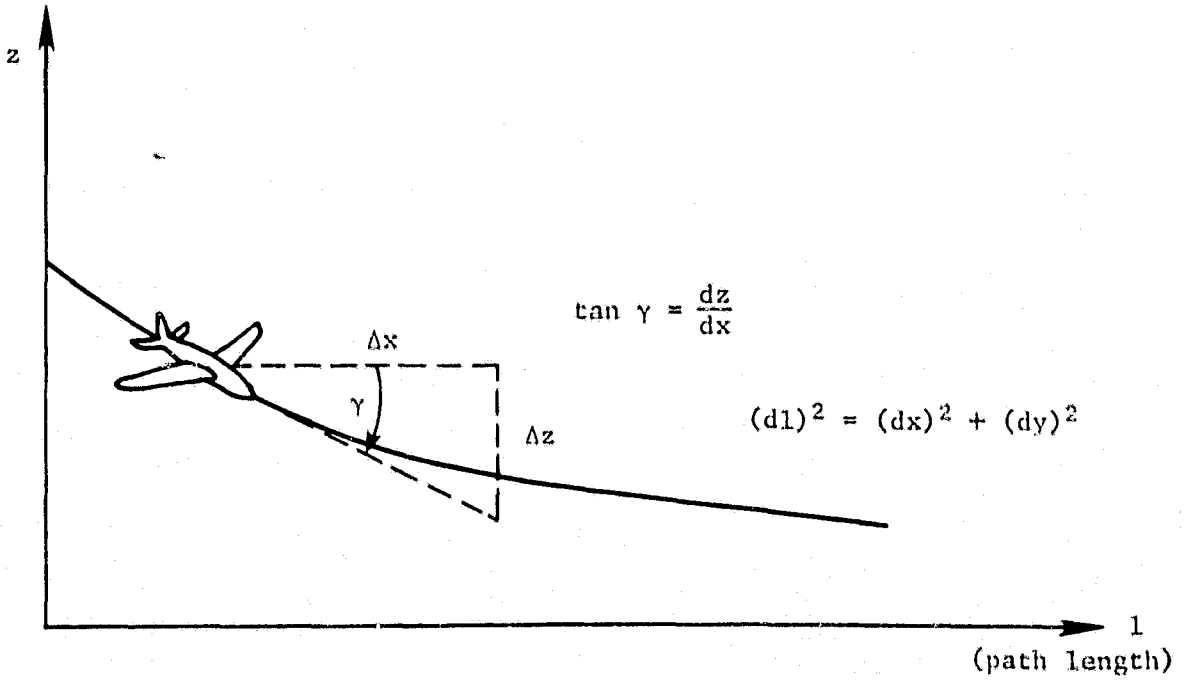


Figure 4.3b Vertical Flight Path Angle

climb, respectively. Referring to Figure (4-3b), it is seen that, in cartesian coordinates

$$\tan \gamma = \frac{dz}{dl} \tag{4-8}$$

so that Eqn. (4-7) may be rewritten

$$\tan \gamma_{d_{max}} \leq \frac{dz}{dl} \leq \tan \gamma_{c_{max}} . \tag{4-9}$$

B. Passenger Ride Quality Constraints

The passenger comfort, or ride quality, constraints limit the operation of an aircraft to those maneuvers found statistically to be acceptable to the passengers. An expression of these limitations (see Ref. [30]) for an aircraft executing a coordinated turn[3] with climb or descent is

$$0.0665|\phi_{max}| + 0.07|p_{max}| + 0.098|\theta_{max}| - 0.118|\delta_{max}| \leq C, \tag{4-10}$$

$$C = \begin{cases} 0.76 \text{ for 90\% passenger satisfaction} \\ 1.76 \text{ for 80\% passenger satisfaction,} \end{cases}$$

where ϕ_{max} = maximum roll angle (degrees)

p_{max} = maximum roll rate (degree/sec)

ϑ_{max} = maximum pitch angle (degrees)

γ_{max} = maximum flight path angle (degrees).

For a coordinated turn and considering only the roll angle part of Eqn. (4-10), the constraint may be written

$$\frac{\bar{V}^2}{R} \leq g\frac{\sin\phi}{\cos\phi} = C_2 g \tag{4-11}$$

---

[3] A coordinated turn is one in which no sideslip occurs, i.e. the component of gravitational force along a wing provides the centripetal force necessary to maintain the particular turning radius being used.

or, using derivatives of the trajectory function:

$$\frac{[1 + (dy/dx)^2]^{3/2}}{d^2y/dx^2} \geq \frac{\bar{V}^2}{C_2 g} \qquad (4\text{-}12)$$

where C_2 = $\begin{cases} 0.19 \text{ for 90\% passenger satisfaction} \\ 0.50 \text{ for 80\% passenger satisfaction} \end{cases}$

$\bar{V}$ = average total velocity

g = acceleration due to gravity

ϕ = roll angle.

C. Regulatory Constraints

Regulations affecting allowable flight paths may pertain either to safety, governmental security, or noise annoyance. Safety restrictions involve single trajectories (e.g. avoidance of low altitude obstacles) and multiple paths (e.g. maintenance of safe separating distances); military installations often prohibit general and commercial aviation flights from passing too closely for security reasons; and there may be certain areas (e.g. schools and hospitals) which are particularly sensitive to noise from nearby aircraft.

Two safety restrictions that apply to every airport have been included in this study. They are:

1. A safe distance should be maintained between any two trajectories as far as is possible. Because several trajectories may share a single runway, the region in which this restriction applies often excludes an area within several miles of the airport. The minimum allowed separating distance used in this work was 800 meters.

2. No landing aircraft should be required to execute a turn of more than 45 degrees, as it crosses the inner marker, in order to come into alignment with the runway. This is referred to here as the final heading requirement.

Restrictions that are related to special areas to be avoided, for security or annoyance reasons, are particular to the given airport community being considered. Although not included in this study, these constraints could be formulated and contained in the problem as described in Chapter VI.

D. Pilot Operating Constraints

It is possible that the optimum trajectories computed would require more maneuvering of an aircraft than a pilot can or is willing to perform, especially in the near terminal area, where his attention is required on a number of matters. The constraints then, are to limit the number and complexity of the turning maneuvers on the flight paths.

E. Threshold Noise Constraint

A difficulty with all three of the annoyance measures discussed in Chapter III (and other ratings) is that an "improvement," as determined by these measures, can be made by exchanging a large number of people exposed to medium noise levels for a small group exposed to extremely high levels. To avoid this possibility, a constraint is imposed that prohibits any area[4] (grid cell) from receiving

---

[4] For some cells near an airport, the high levels may be unavoidable due to runway locations. These cells must be excluded from the restriction.

maximum noise levels ($L_{A,max}$) during a flyover, at or above a "threshold" level, more than N_{max} times per day. The threshold level and value of N_{max} chosen should be as small as possible, without overly restricting the possible trajectories.

Flight Path Model

In selecting an appropriate parametric form for specifying a trajectory, it is necessary to weigh the benefits of a large number of parameters --more degrees of freedom and hence, more flexibility -- against the attendant increase in computation needed to determine the values of those parameters. Chang [26], for example, reported that using only five terms in a Fourier series representation of aircraft landing trajectories (the unknown parameters were the Fourier coefficients) led to overly complicated flight paths, while increasing the number of terms would have resulted in unacceptable computational requirements without necessarily simplifying the paths.

The model chosen for a flight path in this work avoids that problem by directly relating the maximum number of turning maneuvers to the number of parameters to be determined. Each trajectory is represented by a chain of linear segments joined by helical arcs (circular in the ground plane, linear in the profile). Figures 4.4a and 4.4b show an example of a path with three segments. The two intersections of the linear segments in Figure 4.4a are called "corner points;" the cartesian coordinate of these points, along with those of the other trajectories, make up the set of parameters to be determined. In the ground plane, a circular arc with minimum radius (determined by the dynamic and

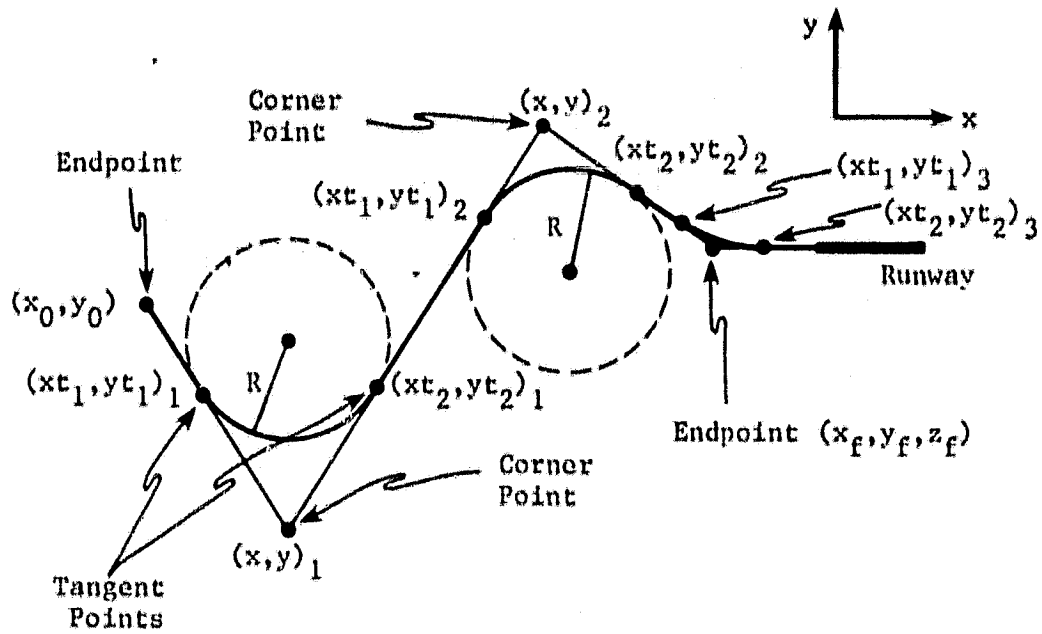


Figure 4.4a Ground Track, Linear Segment Representation

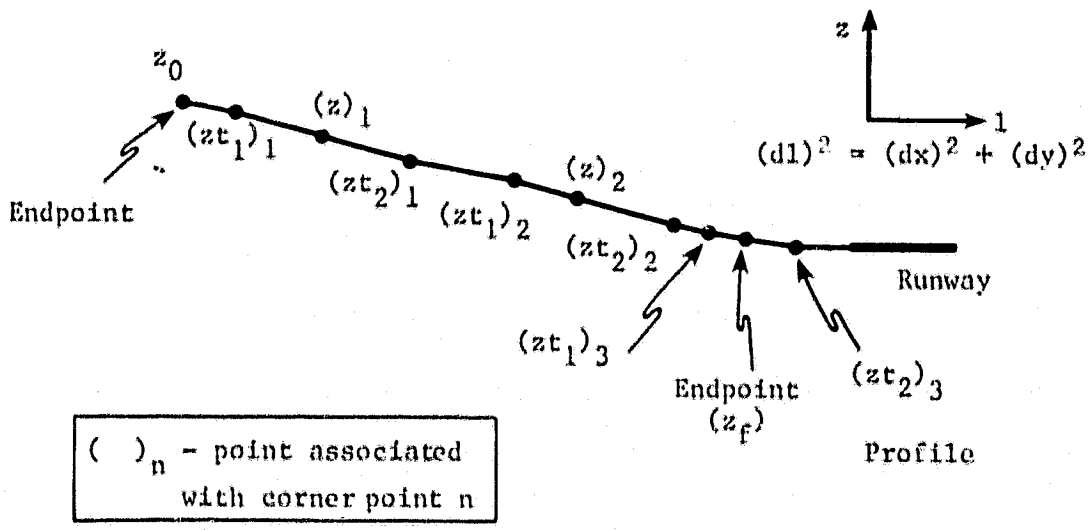


Figure 4.4b Profile, Linear Segment Representation

passenger comfort constraints) is inscribed in the angle between any two linear segments. The arc, tangent to both segments, describes the ground track of the path flown during a turn at that corner. Between tangent points, as shown in Figure 4.4b, the profile of the path is approximated by a linear segment. The resulting turn, in three dimensions, is a helical arc. Since each corner point (requiring three coordinates) may have at most one turn, the complexity of the flight path is limited in a natural way by the allowed flexibility, i.e. the number of corner points used.

Also treated in an implicit way are the endpoint restrictions. Since only the corner points' coordinates are variables, the endpoints of a trajectory may be placed in the appropriate (constant) positions without need of an explicit restriction to hold them constant. For example, in Figure 4.4a, a landing path begins at the entry point (arrival fix) and ends at the inner marker, after which the aircraft flies straight to the runway. For takeoffs (and landings not required to report to the entry point), the endpoint away from the runway may be made variable as shown in Figure 4.5. Here, a pseudo-segment has been added so that the true endpoint becomes a variable corner point. The pseudo-segment must be long enough to prevent a violation of the dynamic and passenger comfort constraints (subsequently discussed), and no noise calculations should involve this fictitious segment.

Explicit statements of the remaining constraints are required for this particular trajectory model. The lateral dynamic restriction, Eqn. (4-6) and passenger comfort constraint Eqn. (4-12) are both inequalities involving the radius of curvature of the ground track; they may be combined to yield

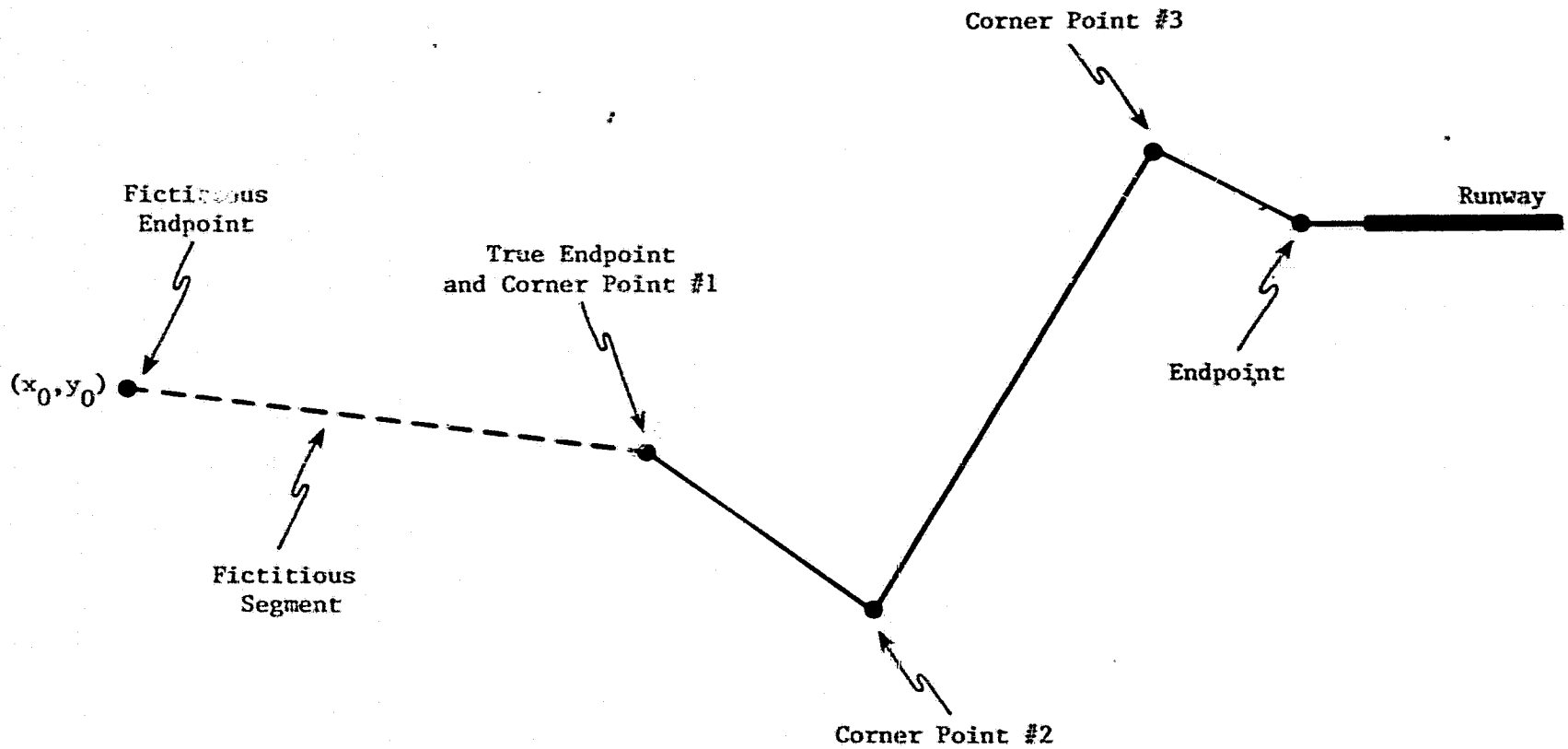


Figure 4.5 Variable Endpoint

$$R \geq \max \left\{ \frac{\bar{V}^2}{C_2 g} , \frac{\bar{V}}{(C_1+C_2C_3)\min(\delta r_1, \delta r_2 \delta r_3)} \right\} , \qquad (4\text{-}13)$$

where R is the radius of curvature of any point on the ground track. For the linear segments of a path, $R \to \infty$; for the turns at the corner point, R is just the radius of the circular arc used to describe a turn in the ground track. Generally, the passenger comfort term (first on the right in Eqn. (4-13)) is the larger, so that a constant value of R (R_{min}) that satisfies Eqn. (4-13) may be employed for all turns once the maximum value of $\bar{V}$ is known for a trajectory.

Incorporation of the longitudinal dynamic constraint is equally straightforward. Eqn. (4-9) involves only the vertical slope (the slope of the profile with respect to the ground plane), a quantity that is readily calculated once the coordinates of the tangent points are known (see Figures 4.4a,b). Appendix C gives a derivation of the formulae used to obtain these coordinates.

Because of the form of the flight path model used here, some additional restrictions on the feasibility of any trajectory must be included. As shown in Figure 4.6, it is possible to specify corner points that result in nonsensical trajectories for a given value of R_{min}. Figure 4.7 indicates the pertinent quantities to use in the restrictions. The first restriction prevents a path from becoming disconnected between any two corner points by requiring that

$$\left| \frac{R_{min}}{\tan\frac{\alpha}{2}} \right| + \left| \frac{R_{min}}{\tan\frac{\beta}{2}} \right| \leq l_2 . \qquad (4\text{-}14)$$

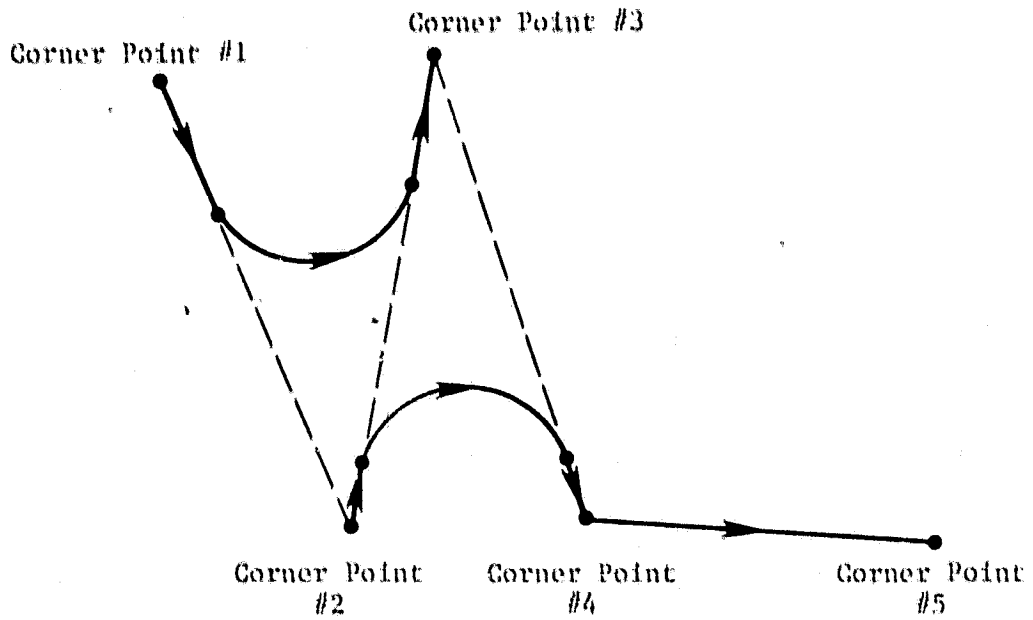


Figure 4.6 Corner Points that Specify an Impossible Trajectory

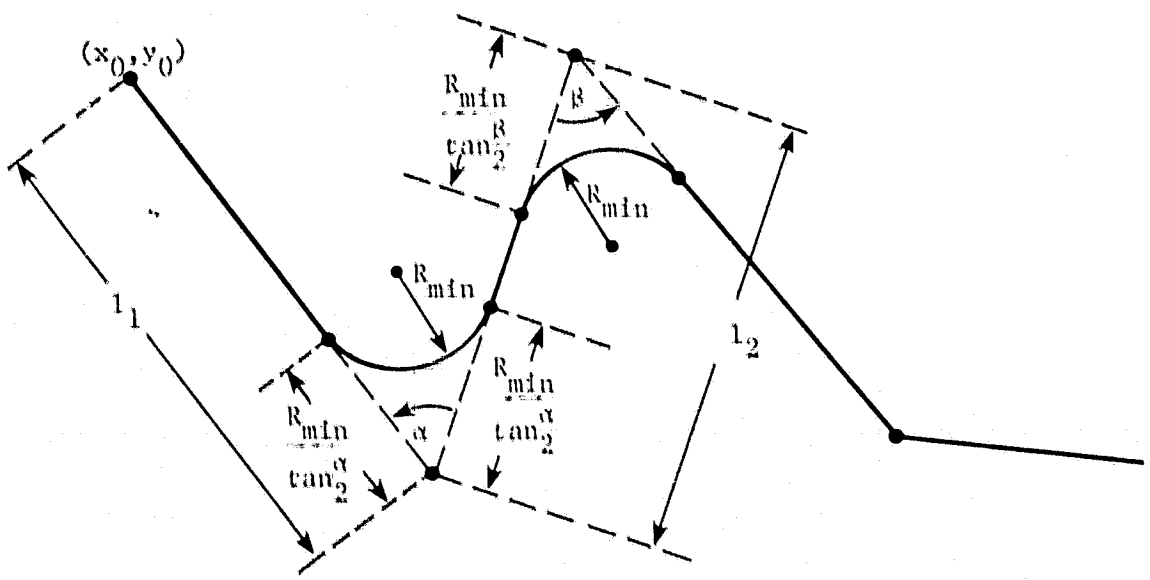


Figure 4.7 Relations Needed to Specify Feasible Flight Paths

Second, for the outermost segment in the chain, an explicit requirement that the circular arc have a tangent point on the segment[5] must be stated, i.e.,

$$\frac{R_{min}}{\tan\frac{\alpha}{2}} \leq l_1. \tag{4-15}$$

Maintaining a safe distance between any two trajectories may be accomplished by sampling the distance between their ground tracks at a number of locations, as shown in Figure 4.8a. This test makes two assumptions: that the profiles of the two paths are sufficiently similar to allow these sampled distances in the ground plane to be good approximations of the actual distances, and that enough samples can be taken to obtain an accurate representation of spacing between the two paths. The first assumption is not an object of concern because it leads, at most, to an underestimation of the spacing. As for the number of samples needed, it has been found in this work that as few as ten measurements will suffice; however, as shown in Figure 4.8b, the successive values $\hat{y}_i$ must be tested not only for minimum magnitude, but also for a change in sign, indicating a crossing of the paths. The two constraints are then:

$$\hat{y}_i \geq d_{min}, \qquad i = 1, s \tag{4-16}$$

$$\hat{y}_i\hat{y}_{i+1} > 0, \qquad i = 1, s-1 \tag{4-17}$$

where s is the number of samples. As stated previously, when two or more flight paths share the same runway, or use runways whose ends

---

[5]The assumption of tangency is necessary for determining the expression for the arc, as developed in Appendix C. On the other segments, Eqn. (4-14) or (4-18), (4-19) assure this condition.

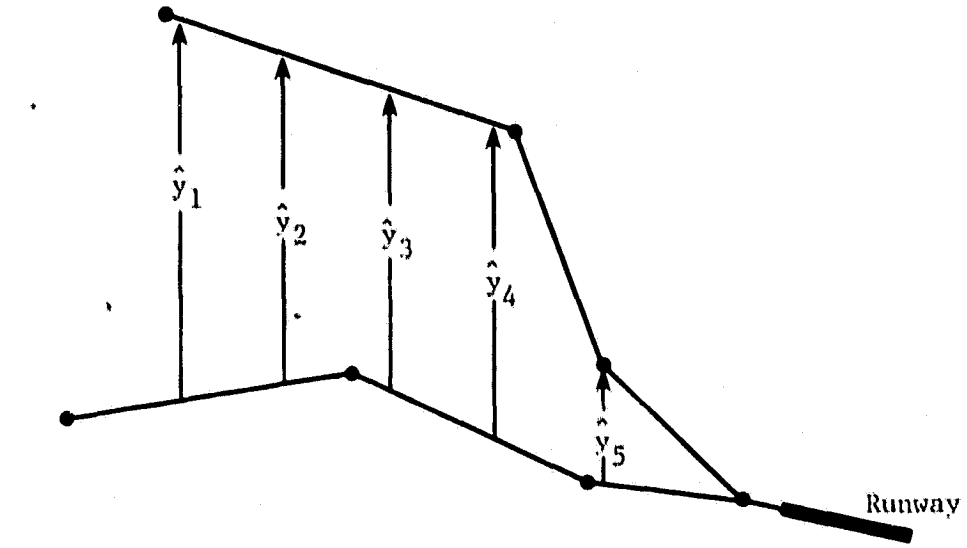


Figure 4.8a Sampling the Distances Between Two Paths

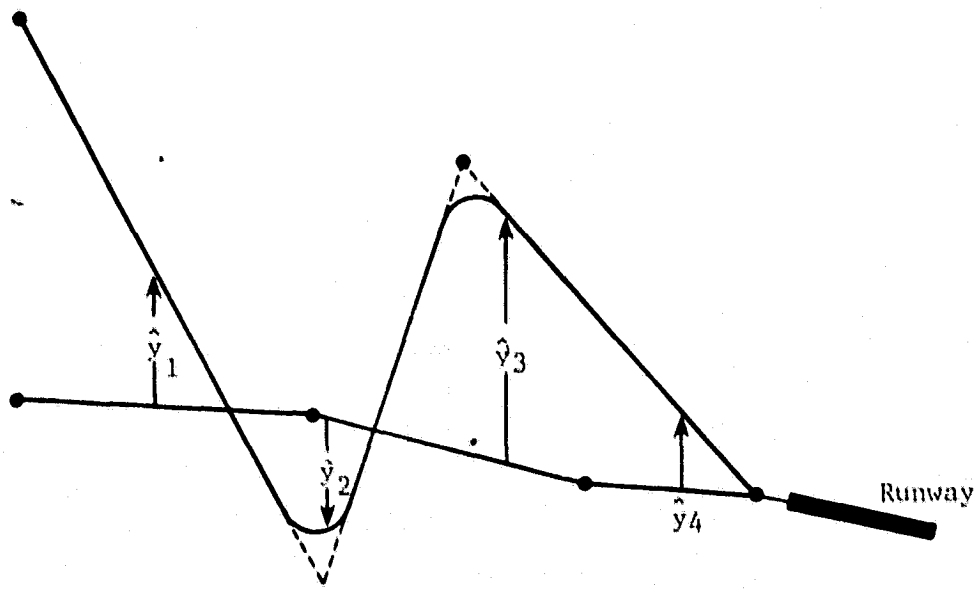


Figure 4.8b Two Crossing Trajectories with $y_i \geq d_{min}$

are close together, the separation constraint, Eqn. (4-16), should not be applied near these runways where the close spacing between trajectories is unavoidable. Further, at larger airports with sophisticated navigational and air traffic control systems, both the separation and crossover restrictions, Eqns. (4-16) and (4-17) may not be required.

The final heading constraint is modelled by requiring that the angle δ, and length l_1, shown in Figure 4.9, satisfy the following:

$$\delta \geq 135°, \tag{4-18}$$

$$l_1 \leq \frac{1}{2}l. \tag{4-19}$$

Together, these guarantee that a landing aircraft will have a path aligned with the runway before reaching it, and that any final turn will be no greater than 45°.

Finally, the threshold noise constraint (which is independent of the trajectory model) requires an analytic form. In each grid cell to which the constraint applies, the maximum A-level ($L_{A,max}$) experienced during each aircraft flyover may not exceed a threshold level ($L_{A,Thresh}$) more than N_{max} times per day. This may be expressed as

$$\sum_{i=N^*}^{N_f} \cdot \max[0,\ L_{A,max_i} - L_{A,Thresh}]_{j\in J} \leq 0 \tag{4-20}$$

where N^* = index of the first flyover after N_{max} flights with $L_{A,max_i} \geq L_{A,Thresh}$

N_f = number of flights during a 24 hour period,

j = index for the grid cells to be included in this constraint, i.e., a set denoted J.

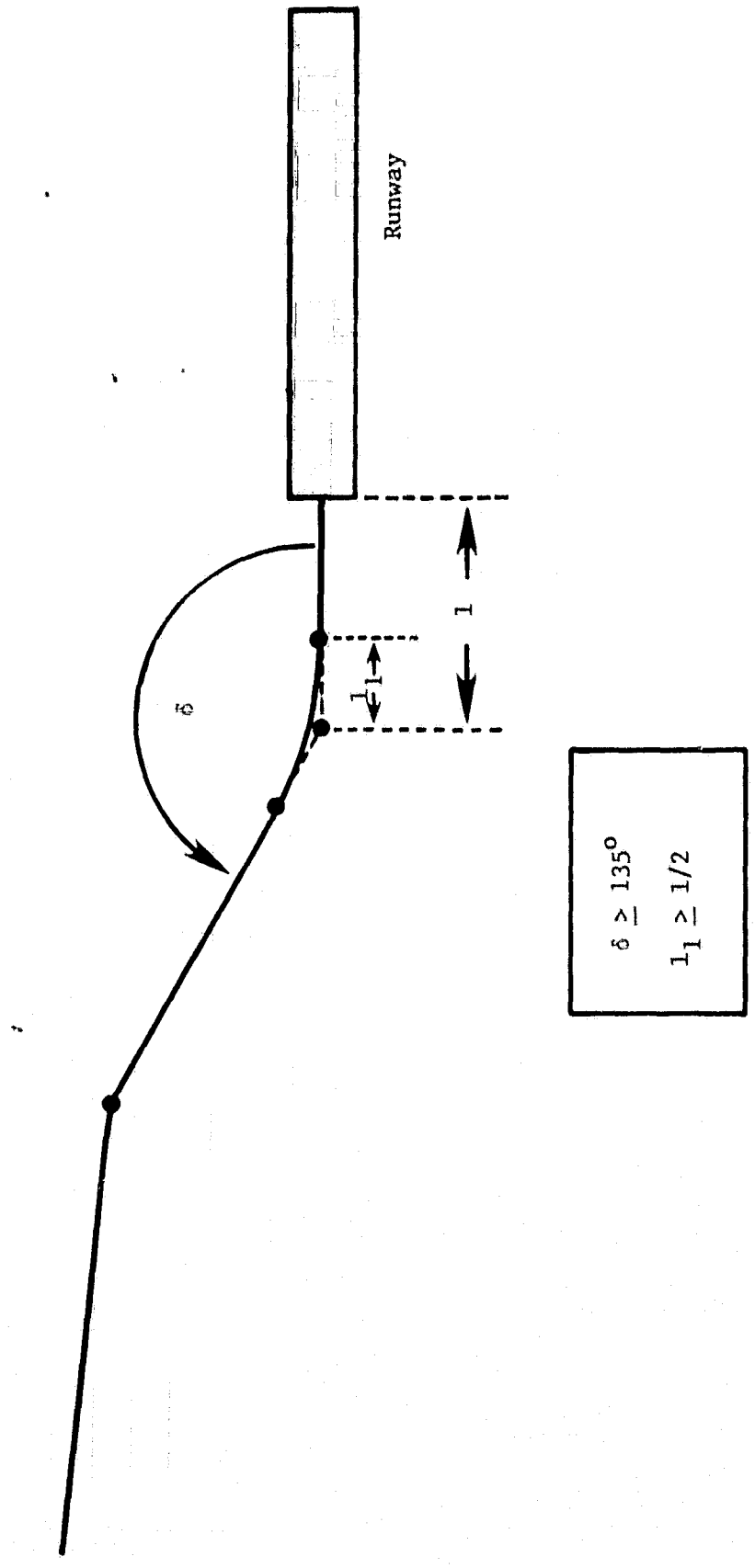


Figure 4.9 The Final Heading Constraints

CHAPTER V

THE OPTIMIZATION SCHEME

In Chapter II, the notion of representing each aircraft trajectory in parametric form was introduced, the idea being to reduce the amount of computation required for determining the optimum set of flight paths. This chapter develops the numerical algorithm that will be used to determing the optimum parameter values. In fact, several algorithms are described in the development, giving a good representation of available methods that are compatible with the solution approach adopted in this work.

All of the optimization methods to be discussed require a criterion, referred to here as the cost function, by which to judge the improvement being made in a system. The cost function must depend, perhaps implicitly, only upon the parameters needed to describe the system (in this case, those that specify the trajectories). As formulated in Chapter VI, the cost will be a function (explicitly) of the noise annoyance, the solution restrictions (constraints), and if desired, other attributes of the system. All of these components, however, depend upon the trajectories, which in turn, are functions of the parameters.

Most optimization algorithms are iterative descent methods. They approximate the solution to a problem usually in several steps, with each successive step making use of previous approximations. This search progresses in such a way that the value of the cost function at each iteration is lower the previous value; the search is then said to be

descending.[1] All of the optimization methods suitable for this work involve derivatives of the cost function with respect to the unknown parameters. Because the cost function includes the annoyance model, which lacks a tractable analytic expression, derivatives of the cost are best determined numerically. This strongly influences the choice of which class of optimization methods to employ.

Of the two suitable for use in this work, primal and approximating methods, the former relies more upon extremely accurate derivative calculations. These methods search over only those parameter combinations not prohibited by the restrictions. Although primal algorithms offer the advantage of "suboptimal" solutions at each iteration[2] (reduced cost and all constraints satisfied), the necessity of large numbers of calculations to compute the derivatives accurately outbalances the attractive features.

The approximating methods begin with a problem in the form

$$\begin{aligned} &\text{minimize } f(\underline{x}) \\ &\underline{x}\epsilon E^n \\ &\text{subject to: } \underline{h}(\underline{x}) \leq \underline{c} \end{aligned} \tag{5-1}$$

where $\underline{x}$ is an n- vector whose components are the parameters to be determined $f(\underline{x})$ is the cost function, $\underline{h}(\underline{x})$ is an m-vector-valued function that includes all the constraints, $\underline{c}$ is a constant m-vector, and E^n

---

[1] It is always possible to cast an optimization problem in a form that seeks a minimum value of the cost.

[2] If for some reason the iterative process must stop before reaching the optimal solution, these sub-optimal solutions offer at least an improvement over the initial condition.

is Euclidean n-space. With the appropriate transformation, the problem becomes unconstrained:[3]

$$\underset{\underline{x}\in E^n}{\text{minimize}}\ \bar{f}(\underline{x}) = f(\underline{x}) + \underline{w}^T\underline{\Phi}(\underline{x}) \qquad (5\text{-}2)$$

where $\underline{w}$ is a m-vector of constants, called the weighting or controlling parameter, and $\underline{\Phi}(\underline{x})$ is an m-vector-valued function that depends upon the problem constraints.

Two forms are popular for use in constructing $\underline{\Phi}(\underline{x})$: barriers and penalties. Barrier functions are not useful in the work for two reasons. First is their requirement that the initial point in the search be a feasible point (satisfy all the constraints). Determining a feasible starting point can be quite difficult for cases with many trajectories having several segments. More important though, is a fundamental defficiency of barrier methods. Theoretically, the term $\underline{w}^T\underline{\Phi}(\underline{x})$ would create a 'barrier" in the n-dimensional surface defined by the modified cost function in Eqn. (5-2). Presumably, the solution point $\underline{x}_k$ (obtained in the k-th iteration) would not move near the barrier, incurring a high cost, in the next iteration. This, however, is not the case if large changes in $\underline{x}$ are allowed; it is possible for the solution to "jump" the barrier, giving an $\underline{x}_{k+1}$ that has a lower cost, but that is infeasible. Without explicitly checking the solution at each iteration, this condition would go "undetected" by the optimization algorithm.

---

[3] Formally, the problem is constrained by the requirement $\underline{x}\in E^n$, but computationally, the solution is unrestricted in real n-space.

Fortunately, the penalty methods are somewhat more robust in this respect. Although they also permit infeasible solutions to pass for feasible ones, the condition is detectable and automatically corrected by the standard procedure for implementing the penalty methods.

A popular form for the penalty function is (see Ref. [23])

$$P_i(\underline{x}) = (\max[0,(h_i(\underline{x})-c_i)])^2, \tag{5-3}$$

with the weighting parameters chosen as

$$\underline{w}^T = [\mu_1, \mu_2, \ldots, \mu_m], \ \mu_i \geq 0. \tag{5-4}$$

The problem is then expressed as (c.f. Eqn. (4-2)):

$$\underset{\underline{x}\in E^n}{\text{minimize}} \ f(\underline{x}) + \sum_{i=1}^{m} \mu_i P_i(\underline{x}). \tag{5-5}$$

To remove totally the approximation in the constraints, introduced by the penalties, the problem must be solved a number of times, with increasing values of the μ_i. A sequence $\{\underline{\mu}^j\}$, with $\mu_i^{j+1} \geq \mu_i^j$, is used to determine a sequence of solutions $\{\underline{x}^j\}$. While some of the $\underline{x}^j$ may violate constraints, it may be shown (see Luenberger [23, pp. 278-280]) that any limit point[4] of the sequence $\{\underline{x}^j\}$ associated with the sequence $\{\underline{\mu}^j\}$ is a solution to the original problem, Eqn. (5-1).

As a practical matter, though, it is necessary only to use a value of $\underline{\mu}$ that results (at worst) in acceptably small violations of the constraints. Some experimentation with the particular problem being solved may be required to determine the components of $\underline{\mu}$ that will give acceptable results. Too large values tend to slow the convergence to the

---

[4] If these exists a subset K of the positive integers such that the subsequence $\{\underline{x}^k\}_{k\in K}$ converges to $\underline{x}$, then $\underline{x}$ is a limit point of $\{\underline{x}^k\}$.

solution, while small values may result in constraint violations that are excessive (by whatever approximation standards apply).

With the contrained problem formulated as an unconstrained one, a suitable optimization method must be selected. The following sections develop several traditional algorithms, leading to the technique chosen for this study.

As with most nonlinear problems, the task of finding the minimum of a nonlinear function $f(\underline{x})$ lacks a general, closed-form solution. Iterative technique, however, can be used to make successive approximations of the solution, until the calculated value is judged to be reasonably close to the actual one.

Once a starting point is chosen, new points are generated by a recursion of the form

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k \tag{5-6}$$

where $\underline{d}_k$ is the "direction" vector and α_k is a scalar that determines how far to move along the direction $\underline{d}_k$ in order to reach $\underline{x}_{k+1}$. The values of α_k and $\underline{d}_k$ vary in the methods subsequently discussed.

Steepest Descent

Perhaps the most "short-sighted" of its type, this method often gives satisfactory results, but more important, it serves as a foundation for the more sophisticated algorithms.

Given a function $f(\underline{x})$ and a point $\underline{x}_0$, the task is to find a good direction (in n-space) in which to search for a new point $\underline{x}_1$, such that $f(\underline{x}_1) \leq f(\underline{x}_0)$. The first approach is conveniently provided by a property of the gradient operator

$$\nabla = \frac{\partial}{\partial x_i} \left[\hat{e}_i\right]^T \qquad \text{(a row vector)}, \qquad (5\text{-}7)$$

namely that $\underline{g}(\underline{x}) = \nabla^T f(\underline{x})$ is a vector that points in the direction in which $f(\underline{x})$ is increasing most rapidly. Since the problem is to find the minimum value of $f(\underline{x})$, the direction in which to search is along $-\underline{g}(\underline{x})$. This will result in the most rapid decrease of $f(\underline{x})$ <u>for that iteration</u>. (Later methods sacrifice this immediate improvement for other advantages.)

As stated earlier, it is not practical to obtain an analytic expression for $\underline{g}(\underline{x})$ in the problem under consideration. An approximate numerical form

$$[\underline{g}(\underline{x})]_i \simeq \left[\frac{f(x_1, x_2, \ldots, x_i + \Delta x, \ldots, x_n) - f(\underline{x})}{\Delta x} \right]_i , \qquad (5\text{-}8)$$

$$i = 1,n$$

has been employed. In the limit $\Delta x \to 0$, Eqn. (5-8) yields the exact gradient; small values of Δx should therefore be used to obtain accurate approximations.

The method of steepest descent is stated as the following algorithm:

<u>Steepest Descent Algorithm</u>

Step 0: Choose an initial point $\underline{x}_0$, set k=0.

Step 1: Compute $\underline{d}_k = -\underline{g}_k(\underline{x}_k)$

Step 2: Minimize (with respect to $\alpha_k > 0$)

$f(\underline{x}_k + \alpha_k \underline{d}_k)$

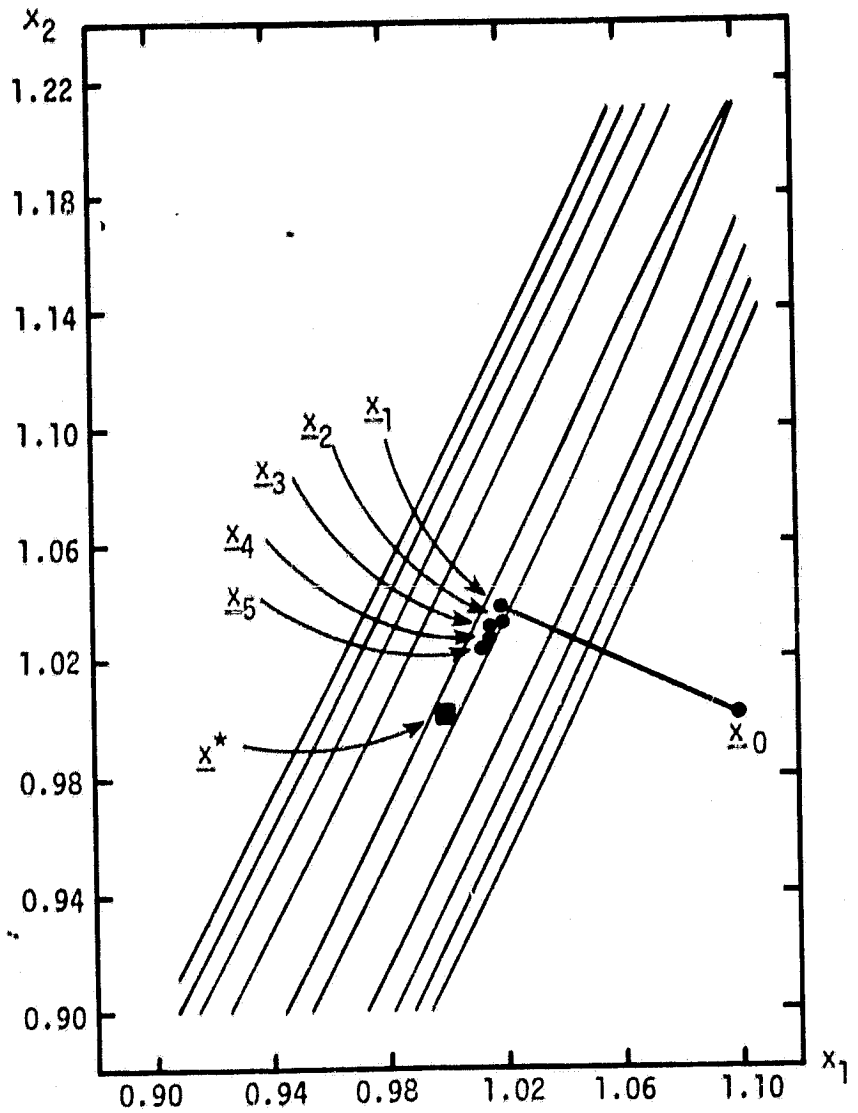


Figure 5.1 Steepest Descent Example

Iterative values of $\underline{x}_k$ up to k=4 are shown as circles, connected by line segments to indicate the progression. This example demonstrates a major problem of steepest descent, namely, its slow convergence properties. Its initial step to $\underline{x}_1$ is directly "downhill," causing a large decrease in the value of $f(\underline{x})$; however, once in the shallow region, it oscillates ineffectually, making slow progress. The curvature along the valley prevents $-\underline{g}(\underline{x})$, the search direction, from pointing directly to the solution. A more rigorous explanation of the convergence properties of this algorithm may be found in Luenberger [23, pp. 148-155].

Newton's Method

A more sophisticated approach to the general problem of minimizing a function f is to approximate that function with a quadratic form in the region of the solution:

$$f(\underline{x}) \simeq c - \underline{b}^T\underline{x} + \tfrac{1}{2}\,\underline{x}^T\underline{Q}\underline{x}. \tag{5-10}$$

This approximate function has a minimum at

$$\underline{x}^* = \underline{Q}^{-1}\underline{b} \tag{5-11}$$

provided that $\underline{Q}$ is a positive definite symmetric matrix.[5] Taylor's Theorem[6] may be used to expand $f(\underline{x})$ about $\underline{x}_k$ keeping only up to quadratic terms:

$$f(\underline{x}) \simeq f(\underline{x}_k) + \nabla f(\underline{x}_k)(\underline{x}-\underline{x}_k) + \tfrac{1}{2}(\underline{x}-\underline{x}_k)^T\,\underline{F}(\underline{x}_k)(\underline{x}-\underline{x}_k), \tag{5-12}$$

where $\underline{F}(\underline{x}_k)$ the Hessian of f, is the nxn matrix of second-partial derivatives of $f(\underline{x})$, evaluated at $\underline{x}_k$.

---

[5] In R^n, a matrix $\underline{Q}$, is positive definite if $\underline{x}^T\underline{Q}\underline{x} > 0$ for all $\underline{x}\neq\underline{0}$.

[6] See Pierre [32].

$$\underline{F}(\underline{x}) = \left[\frac{\partial^2 f(\underline{x})}{\partial x_i \partial x_j}\right] . \tag{5-13}$$

Comparing terms of (5-12) with (5-11) gives a minimum[7] at

$$(\underline{x}-\underline{x}_k) = -[\underline{F}(\underline{x}_k)]^{-1} \quad \underline{\quad} f(\underline{x}_k)^T, \tag{5-14}$$

from which follows the recursion

$$\underline{x}_{k+1} = \underline{x}_k - [\underline{F}(\underline{x}_k)]^{-1} \quad \underline{\quad} f(\underline{x}_k)^T. \tag{5-15}$$

The algorithm for Newton's method is then:

Newton's Method

Step 0: Choose an initial point $\underline{x}_0$, set k = 0.

Step 1: Compute $\underline{g}(\underline{x}_k)$, $[\underline{F}(x_k)]^{-1}$.

Step 2: Set $\underline{x}_{k+1} = \underline{x}_k - [\underline{F}(\underline{x}_k)]^{-1}\underline{g}(\underline{x}_k)$.

Set k=k+1

Go to Step 1.

As before, this procedure continues until a stopping criterion is met.

No line search is required, but computing, inverting, and storing the inverse Hessian requires more computing resource than does the steepest descent algorithm. Newton's method does, however, converge rapidly when $\underline{x}_k$ is near the solution $\underline{x}^*$. This depends somewhat upon how nearly quadratic $f(\underline{x})$ is: a scaling factor α_k, and a line search may be included for cases where the quadratic approximation is not sufficient. The recursion then becomes

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k[\underline{F}(\underline{x}_k)]^{-1}\underline{g}(\underline{x}_k), \tag{5-16}$$

---

[7] The requirement that $\underline{Q}$ (and $\underline{F}$) be positive definite is based upon the necessary and sufficient conditions for $\underline{x}^*$ to be a minimum point. (See Luenberger [23, pp. 110-114]).

and a line search to minimize $f(\underline{x}_k + \alpha_k[\underline{F}(\underline{x}_k)]^{-1}\underline{g}(\underline{x}_k))$ must be added to Step 2 of the algorithm.

Newton's Method: Example 2

Figure 5.2 shows the results of applying Newton's method to the same problem as in Example 1. The pure form of the method, without the search parameter α_k, was used with excellent results. The seemingly odd behavior exhibited in iteration 1 demonstrates how this method, in sacrificing the initial rapid progress of steepest descent, converges swiftly on the solution after only five iterations.

Conjugate Gradient Methods

Of the descent techniques examined thus far, one converges quite slowly when near the solution, and the other, quite rapidly, but at the expense of large increases in computation. A good compromise is provided by the conjugate gradient methods. These are developed qualitatively here; theoretical treatments may be found in Ref. [23] and [32].

Conjugate gradient techniques are a subset of the more general conjugate direction method. As the name implies, gradient information is used in computing the search directions; it is the special conjugacy of these directions that distinguishes the methods. The quadratic problem considered previously serves to illustrate the notions involved.

A quadratic function

$$f(\underline{x}) = \tfrac{1}{2}\underline{x}^T\underline{Q}\underline{x} - \underline{b}^T\underline{x} \qquad (5\text{-}17)$$

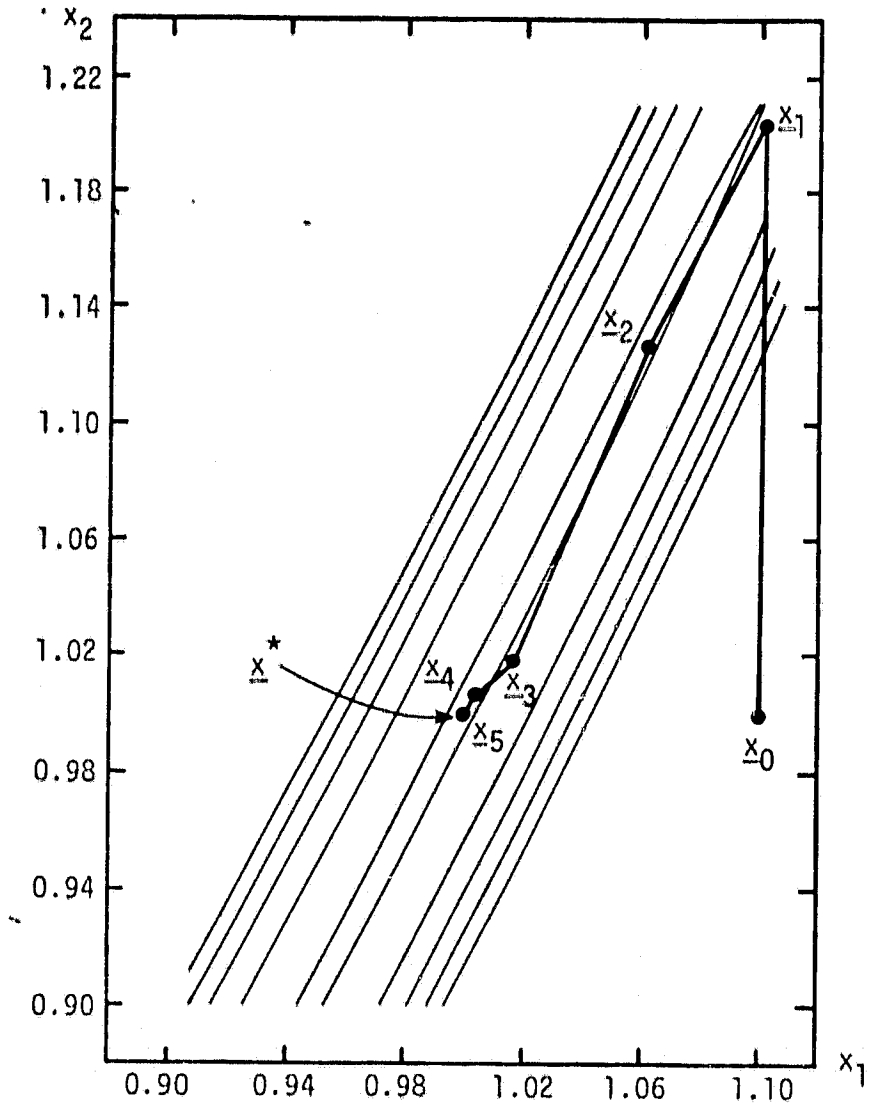


Figure 5.2 Newton's Method Example

has a minimum at

$$\underline{x}^* = \underline{Q}^{-1}\underline{b} \tag{5-18}$$

if $\underline{Q}$ is positive definite. This solution may be expanded, using any linearly independent set of vectors $\{\underline{d}_i\}_{i=0}^{n-1}$, as

$$\underline{x}^* = \sum_{i=0}^{n-1} \alpha_i \underline{d}_i . \tag{5-19}$$

If the $\underline{d}_i$ are $\underline{Q}$-orthogonal (or $\underline{Q}$-conjugate), i.e. $\underline{d}_j^T \underline{Q} \underline{d}_i = 0$ for $i \neq j$, then the following results[8]:

$$\underline{Q}^{-1}\underline{b} = \sum_{i=0}^{n-1} \alpha_i \underline{d}_i , \tag{5-20}$$

$$\underline{Q}\underline{Q}^{-1}\underline{b} = \sum_{i=0}^{n-1} \alpha_i \underline{Q}\underline{d}_i , \tag{5-21}$$

$$\underline{d}_j^T \underline{Q}\underline{Q}^{-1}\underline{b} = \sum_{i=0}^{n-1} \alpha_i \underline{d}_j^T \underline{Q}\underline{d}_i , \tag{5-22}$$

$$\underline{d}_j^T \underline{b} = \alpha_j \underline{d}_j^T \underline{Q}\underline{d}_j , \tag{5-23}$$

$$\alpha_j = \frac{\underline{d}_j \underline{b}}{\underline{d}_j^T \underline{Q}\underline{d}_j} . \tag{5-24}$$

The solution, in terms of the conjugate directions, is then

$$\underline{x}^* = \sum_{i=0}^{n-1} \frac{\underline{d}_i^T \underline{b}}{\underline{d}_i^T \underline{Q}\underline{d}_i} \underline{d}_i . \tag{5-25}$$

---

[8] Proof that the $\underline{Q}$-conjugate set $\{\underline{d}_i\}$ are linearly independent:

Assume that the $\underline{d}_i$ are not linearly independent. Then there exist $\alpha_i \neq 0$ such that $\sum_{i=0}^{n-1} \alpha_i \underline{d}_i = \underline{0}$. Multiplying by $\underline{d}_j \underline{Q}$ yields (for each i) $\alpha_i \underline{d}_i \underline{Q}\underline{d}_i = 0$. Since $\underline{Q}$ is positive definite, $\underline{d}_i \underline{Q}\underline{d}_i > 0$ and $\alpha_i \equiv 0$, contradicting the assumption.

All that remains is to specify the $\underline{d}_i$ in terms of $\underline{Q}$ and $\underline{b}$, the known quantities in the problem.

What the conjugate gradient algorithm does is to compute the $\underline{d}_i$ iteratively, using linear combinations of the gradients in all previous iterations.

Conjugate Gradient Algorithm

Step 0: Set $k = 0$, $\underline{x}_k = \underline{x}_0$, $\underline{d}_0 = -\underline{g}_0 = \underline{b} - \underline{Q}\underline{x}_0$.

Step 1: Compute $\alpha_k = \dfrac{-\underline{g}_k^T \underline{d}_k}{\underline{d}_k^T \underline{Q}_k \underline{d}_k}$ (5-26)

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k \tag{5-27}$$

Step 2: $\beta_k = \dfrac{\underline{g}_{k+1}^T \underline{Q}\underline{d}_k}{\underline{d}_k^T \underline{Q}_k \underline{d}_k}$ (5-28)

$$\underline{d}_{k+1} = -\underline{g}_{k+1} + \beta_k \underline{d}_k \tag{5-29}$$

$$(\underline{g}_k = \underline{Q}\underline{x}_k - \underline{b}) \tag{5-30}$$

Step 3: Set $k = k+1$ and go to Step 1.

These iterative calculations are exact for a purely quadratic problem and the solution, Eqn. (5-25) is obtained in n steps.

In general, the function being minimized is not quadratic. Near a minimum point, the Taylor expansion is

$$f(\underline{x}) = f(\underline{x}^*) + \nabla f(\underline{x}^*)(\underline{x}-\underline{x}^*) + \tfrac{1}{2}(\underline{x}-\underline{x}^*)^T \underline{F}(\underline{x}^*)(\underline{x}-\underline{x}^*) + 0(|\underline{x}-\underline{x}^*|^3). \tag{5-31}$$

A necessary condition for $\underline{x}^*$ to be an unconstrained minimum point is that $\nabla f(\underline{x}^*) \equiv \underline{0}$. The fouth term becomes negligible as $\underline{x} \to \underline{x}^*$, leaving only the quadratic term in $\underline{x}$. This leads to the following extension of the conjugate gradient algorithm in which $\underline{Q}$ is replaced by $\underline{F}$.

General Conjugate Gradient Algorithm

Step 0: Set $k = 0$, $\underline{d}_0 = -\underline{g}_0 = \nabla^T f(\underline{x}_0)$ (5-31)

Step 1: Compute $\alpha_k = \dfrac{-\underline{g}_k^T \underline{d}_k}{\underline{d}_k^T \underline{F}(\underline{x}_k) \underline{d}_k}$ (5-32)

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k \quad (5\text{-}33)$$

Step 2: Compute $\underline{g}_{k+1} = \nabla^T f(\underline{x}_{k+1})$ (5-34)

If $k \neq mn-1$ (m an integer),

compute $\beta_k = \dfrac{\underline{g}_{k+1}^T \underline{F}(\underline{x}_k) \underline{d}_k}{\underline{d}_k^T \underline{F}(\underline{x}_k) \underline{d}_k}$ (5-35)

set $\underline{d}_{k+1} = -\underline{g}_{k+1} + \beta_k \underline{d}_k$ (5-36)

$k = k+1$

Else,

Set $\underline{d}_{k+1} = -\underline{g}_{k+1}$ (5-37)

$k = k+1$

Go to Step 1.

In Step 2, the "else" branch is called a spacer (or restart) step. It is inserted to aid in the convergence to the solution by reinitiating the conjugate direction calculations every n steps.

A disadvantage of the above algorithm is its requirement of calculating $\underline{F}(\underline{x})$. Even if storage of the nxn matric is not an obstacle, numerical accuracy (along with that of $\underline{g}(\underline{x})$) is still a concern. Errors may grow during the iterative construction of the $\underline{d}_k$ to the point that even the spacer step cannot "recover" the search and no progress is made.

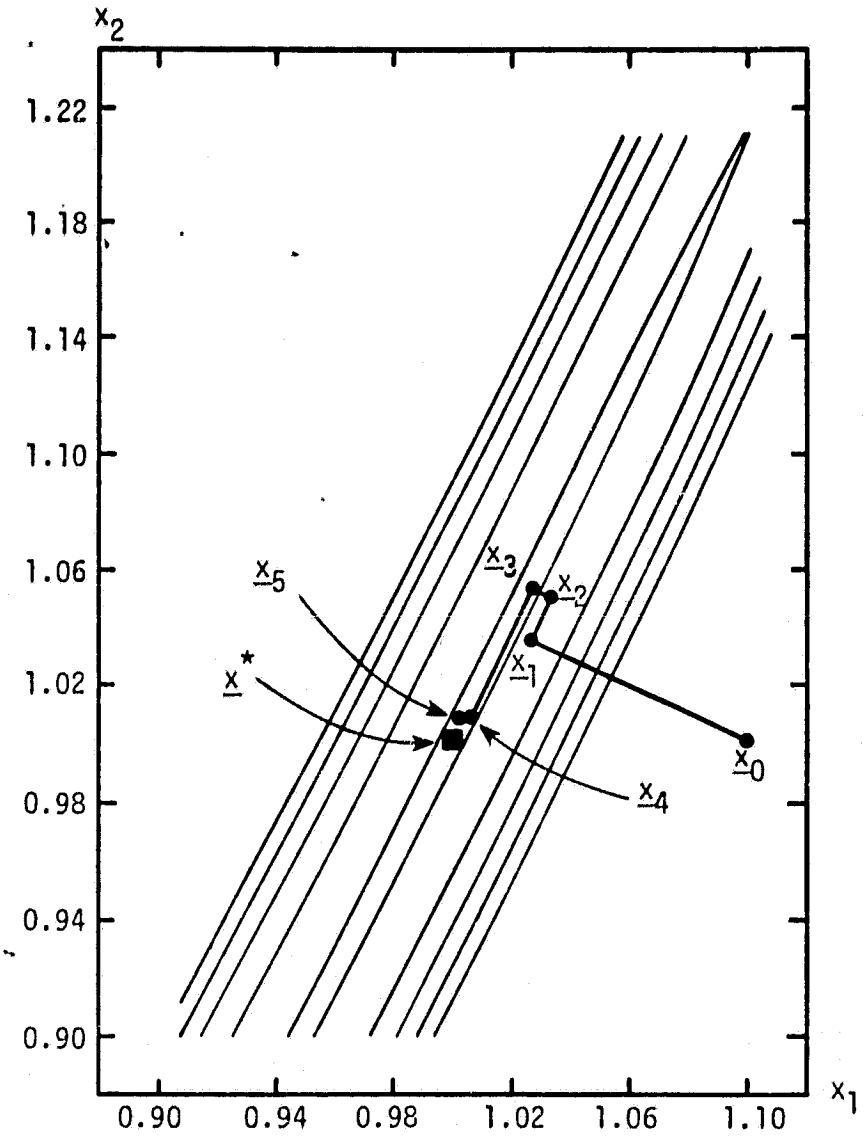


Figure 5.3 Conjugate Gradients Example

Step 2: Set $\underline{d}_k = -\underline{S}_k \underline{g}_k$. (5-40)

Step 3: Minimize $f(\underline{x}_k + \alpha_k \underline{d}_k)$ with respect to $\alpha \geq 0$ (5-41)

Set $\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k$ (5-42)

$\underline{p}_k = \alpha_k \underline{d}_k$ (5-43)

$\underline{q}_k = \underline{g}_{k+1} - \underline{g}_k$ (5-44)

(Note: α_k must be selected so that $\underline{p}_k^T \underline{q}_k > 0$)

Step 4: If $k \neq mn$ (m an integer), set

$$\underline{S}_{k+1} = \left[\underline{S}_k - \frac{\underline{S}_k \underline{q}_k^T \underline{q}_k \underline{S}_k}{\underline{q}_k^T \underline{S}_k \underline{q}_k} \right] \frac{\underline{p}_k^T \underline{q}_k}{\underline{q}_k \underline{S}_k \underline{q}_k} + \frac{\underline{p}_k \underline{p}_k^T}{\underline{p}_k^T \underline{q}_k} \quad . \tag{5-45}$$

Set k = k+1.

Go to Step 2.

Else,

Go to Step 1.

A. Requirement for step-wise descent

The above algorithm, along with steepest descent and Newton's method, belongs to a class of methods that use the recursion

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k \underline{M}_k \underline{g}_k \tag{5-46}$$

to compute the minimum of $f(\underline{x})$, where $\alpha \geq 0$, and $\underline{M}_k$ is an n x n matrix that modifies the direction $-\underline{g}_k$. For steepest descent, $\underline{M}_k \bar{\nabla} \underline{I}$, for Newton's Method, $\underline{M}_k = [\underline{F}(\underline{x}_k)]^{-1}$, and for this method, $\underline{M}_k = \underline{S}_k \simeq [\underline{F}(\underline{x}_k)]^{-1}$ (a point which remains to be proved). A desirable property of such iterative methods is that each step produces a lower value of $f(\underline{x})$. The requirement on $\underline{M}_k$ that insures this is now derived.

Using Taylor's theorem to expand $f(\underline{x})_{\underline{x}=\underline{x}_{k+1}}$ about $\underline{x}_k$ gives:

$$f(\underline{x}_{k+1}) = f(\underline{x}_k) + \nabla f(\underline{x}_k)(\underline{x}_{k+1} - \underline{x}_k) + 0(|\underline{x}_{k+1} - \underline{x}_k|^2), \tag{5-47}$$

into which Eqn. (5-46) may be inserted to yield:

$$f(\underline{x}_{k+1}) = f(\underline{x}_k) - \alpha_k \underline{g}_k^T \underline{M}_k \underline{g}_k + 0(1\alpha 1^2). \tag{5-48}$$

For small α, the trailing term becomes negligible compared to the second term on the right. If $f(\underline{x}_{k+1})$ is to be less than $f(\underline{x}_k)$ for small $\alpha_k > 0$, then

$$- \underline{g}_k^T \underline{M}_k \underline{g}_k > 0. \tag{5-49}$$

This can be insured for all $\underline{g}_k \neq \underline{0}$ by requiring the $\underline{M}_k$ be positive definite. (Clearly, for steepest descent this is the case, but for Newton's method, $\underline{M}_k = [\underline{F}(\underline{x}_k)]^{-1}$ is not guaranteed to be positive definite except near a minimum.)

B. Positive Definiteness of $\underline{S}_k$

Given the algorithm for generating the $\underline{S}_k$, it will be shown that the positive definiteness of $\underline{S}_{k+1}$ follows from that of $\underline{S}_k$. For $\underline{x} \in E^n$,

$$\underline{x}^T \underline{S}_{k+1} \underline{x} = \gamma_k \underline{x}^T \underline{S}_k \underline{x} - \gamma_k \frac{(\underline{x}^T \underline{S}_k \underline{q}_k)^2}{\underline{q}_k^T \underline{S}_k \underline{q}_k} + \frac{(\underline{x}^T \underline{p}_k)^2}{\underline{p}_k^T \underline{q}_k} \tag{5-50}$$

where

$$\gamma_k = \frac{\underline{p}_k^T \underline{q}_k}{\underline{q}_k^T \underline{S}_k \underline{q}_k} .$$

Substituting $\underline{a} = \underline{S}_k^{\frac{1}{2}} \underline{p}_k$, $\underline{b} = \underline{S}^{\frac{1}{2}} \underline{q}_k$ gives

$$\underline{x}^T \underline{S}_{k+1} \underline{x} = \gamma_k \left[\frac{(\underline{a}^T \underline{a})(\underline{b}^T \underline{b}) - (\underline{a}^T \underline{b})^2}{\underline{b}^T \underline{b}} \right] + \frac{(\underline{x}^T \underline{p}_k)^2}{\underline{p}_k^T \underline{q}_k} . \tag{5-51}$$

By definition, $\underline{q}_k = \underline{g}_{k+1} - \underline{g}_k$, so that

$$\underline{p}_k^T \underline{q}_k = \underline{p}_k^T \underline{g}_{k+1} - \underline{p}_k^T \underline{g}_k . \tag{5-52}$$

Since $\underline{x}_{k+1}$ is the minimum of $f(\underline{x})$ along $\underline{p}_k = \alpha_k \underline{d}_k$, then

$$\frac{\partial f}{\partial \alpha}(\underline{x}_k + \alpha \underline{d}_k) = \frac{\nabla f(\underline{x}_{k+1})\underline{d}_k}{\underline{d}_k^T \underline{d}_k} = \frac{\alpha_k \underline{g}_{k+1}^T \underline{p}_k}{\underline{d}_k^T \underline{d}_k} = 0 \qquad (5\text{-}53)$$

Therefore Eqn. (5-52) yields

$$\underline{p}_k^T \underline{q}_k = -\underline{p}_k^T \underline{g}_k \qquad (5\text{-}54)$$

By definition, $\underline{p}_k = \alpha_k \underline{d}_k = -\alpha_k \underline{S}_k \underline{g}_k$, so that

$$\underline{p}_k^T \underline{q}_k = \alpha_k (\underline{S}_k \underline{g}_k)^T \underline{g}_k = \alpha_k \underline{g}_k^T \underline{S}_k \underline{g}_k. \qquad (5\text{-}55)$$

Eqn. (5-51) may now be written

$$\underline{x}^T \underline{S}_{k+1} \underline{x} = \gamma_k \left[\frac{(\underline{a}^T \underline{a})(\underline{b}^T \underline{b}) - (\underline{a}^T \underline{b})^2}{\underline{b}^T \underline{b}} \right] + \frac{(\underline{x}^T \underline{p}_k)^2}{\alpha_k \underline{g}_k^T \underline{S}_k \underline{g}_k} \qquad (5\text{-}56)$$

By the Cauchy-Schwarz inequality,[9] the first term on the right of Eqn. (5-56) is positive semi-definite, as is the second term, by virtue of the definition of inner product. To demonstrate that both terms cannot vanish simultaneously, assume that $\underline{a} = \xi \underline{b}$, for which case the first term vanishes. Then $\underline{x} = \xi \underline{q}_k$ and

$$\underline{p}_k^T \underline{x} = \xi \underline{p}_k^T \underline{q}_k = \xi \alpha_k \underline{g}_k^T \underline{S}_k \underline{g}_k \neq 0 \qquad (5\text{-}57)$$

since $\underline{S}_k$ is positive definite. Therefore, $\underline{x}^T \underline{S}_{k+1} \underline{x} > 0$ for all $\underline{x} \neq 0$. Since the algorithm starts with $\underline{S}_0 = \underline{I}$, all of the $\underline{S}_k$ will be positive definite. Q.E.D.

An extremely useful piece of information in this proof regards the determination of α_k. The magnitude of $\underline{p}_k^T \underline{q}_k$ in Eqn. (5-55) is not paramount; only its sign affects the definiteness of $\underline{S}_{k+1}$. For this

---

[9] Cauchy-Schwarz Inequality.
$(\underline{a}^T \underline{a})(\underline{b}^T \underline{b}) \geq (\underline{a}^T \underline{b})^2$

reason, the algorithm requires that the line search in Step 3 be accurate enough only to give $\underline{p}_k^T \underline{g}_k > 0$. (When applied to nonquadratic problems, the requirement that α_k minimizes $f(\underline{x})$ along $\underline{x}_k + \alpha_k \underline{d}_k$ becomes more stringent).

C. Convergence of $\underline{S}_k$ to $\underline{F}^{-1}$

For the modified Davidon-Fletcher-Powell method, it may be shown that if $f(\underline{x})$ is quadratic ($\underline{F}$ constant and positive definite), then

$$\underline{p}_i^T \underline{F} \underline{p}_j = 0, \qquad 0 \leq i < j \leq k \tag{5-58}$$

and

$$\underline{S}_{k+1} \underline{F} \underline{p}_i = \underline{p}_i, \quad 0 \leq i \leq k, \tag{5-59}$$

from which the convergence of the algorithm in a finite number of steps will be proved, along with the fact that $\underline{S}_n = \underline{F}^{-1}$.

A Taylor expansion of $\underline{g}(\underline{x}) = \nabla^T f(\underline{x})$ gives

$$\underline{g}(\underline{x}) \simeq \underline{g}(\underline{x}_k) + \underline{F}(\underline{x} - \underline{x}_k). \tag{5-60}$$

Using Eqn. (5-42) - (5-44) in the above yields

$$\underline{g}_k = \underline{g}_{k+1} - \underline{g}_k = \underline{F}\underline{x}_{k+1} - \underline{F}\underline{x}_k = \underline{F}\underline{p}_k. \tag{5-61}$$

From (5-45)

$$\underline{S}_{k+1} \underline{F} \underline{p}_k = \underline{S}_{k+1} \underline{g}_k = \underline{p}_k \tag{5-62}$$

The proof of (5-58) and (5-59) is by induction: They are true for k = 0 by virtue of (5-61) and (5-62). Assuming that they are true for iteration k-1, it will be shown that they are true for iteration k.

Using (5-61), $\underline{g}_k$ may be constructed from previous iterations:

$$\underline{g}_k = \underline{g}_{i+1} + \underline{F}(\underline{p}_{i+1} + \ldots + \underline{p}_{k-1}) \tag{5-63}$$

From (5-53) and (5-58)

$$\underline{p}_i^T \underline{g}_k = \underline{p}_i^T \underline{g}_{i+1} = 0 \ , \quad 0 \leq i < k \ . \tag{5-64}$$

Transposing (5-62) and substituting into (5-64) yields

$$\underline{p}_i^T \underline{F} \underline{S}_k \underline{g}_k = 0 \ , \quad i < k. \tag{5-65}$$

From the definition of $\underline{p}_k$, the above may be rewritten

$$\underline{p}_i^T \underline{F} \underline{p}_k = 0 \ , \quad i < k, \tag{5-66}$$

proving (5-58) for k.

Still assuming that (5-61) is true for k-1,

$$\underline{S}_k \underline{F} \underline{p}_i = \underline{p}_i \ , \quad 0 \leq i < k, \tag{5-67}$$

which combined with (5-62) gives

$$\underline{S}_{k+1} \underline{F} \underline{p}_i = \underline{p}_i \ , \quad 0 \leq i \leq k. \qquad \text{Q.E.D.}$$

Since the $\underline{p}_k$ are $\underline{F}$-orthogonal, they are linearly independent (see footnote 8)., If the matrix $\underline{P}$ whose columns are the $\underline{p}_k$ is constructed

$$\underline{P} = [\underline{p}_0 \mid \underline{p}_2 \mid \cdots \mid \underline{p}_{n-1}] \ , \tag{5-68}$$

then (5-59) yields

$$\underline{S}_n \ \underline{F} \ \underline{P} = \underline{P} \ . \tag{5-69}$$

The linear independence of the $\underline{p}_k$ insures that $\underline{P}^{-1}$ exists, so that

$$\underline{S}_n \ \underline{F} = \underline{P}\underline{P}^{-1} = \underline{I} \tag{5-70}$$

and

$$\underline{S}_n = \underline{F}^{-1} \ . \tag{5-71}$$

Finally, this method is seen to be a conjugate direction method, by virtue of the $\underline{F}$-conjugacy of the $\underline{p}_k$. For the quadratic case used in the preceeding analyses, it will converge in n steps or less. The presence of the scaling factor γ_k alters the eigenvalues in $\underline{S}_k \simeq \underline{F}^{-1}$ at each iteration such that they span a small range. This leads to a

better rate of convergence than was generally obtained in the original method. For a development of this feature, see Ref. [23]. A pure steepest descent step every n steps insures that the conjugate direction process will be initiated in the vicinity of the minimum, where its convergence is fastest.

Modified D-F-P Method: Example 4

Returning to the problem (5-9) used in the previous examples, the last algorithm is employed with excellent results. Figure 5.4 shows that after five iterations, the approximate solution is comparable to that of the conjugate gradient method in Example 3.

The last algorithm was chosen for use in this study for several reasons. First, it has convergence properties superior to those of steepest descent. Second, it requires only first order derivative information, which may be generated practically. Finally, the search directions employed are guaranteed to decrease the cost function at each iteration, an assurance that the conjugate direction methods lack for nonquadratic problems.

Until now, the problem of nonconvex (or multimodal) cost functions has not been considered. The preceeding search methods all refer to finding "a minimum" of the cost, but in nonconvex cases, there is more than one minimum. The lowest valued of these so-called relative or local minima is referred to as the absolute or global minimum. There is no way to guarantee that any of the standard iterative techniques will converge to the global minimum of a nonconvex function.

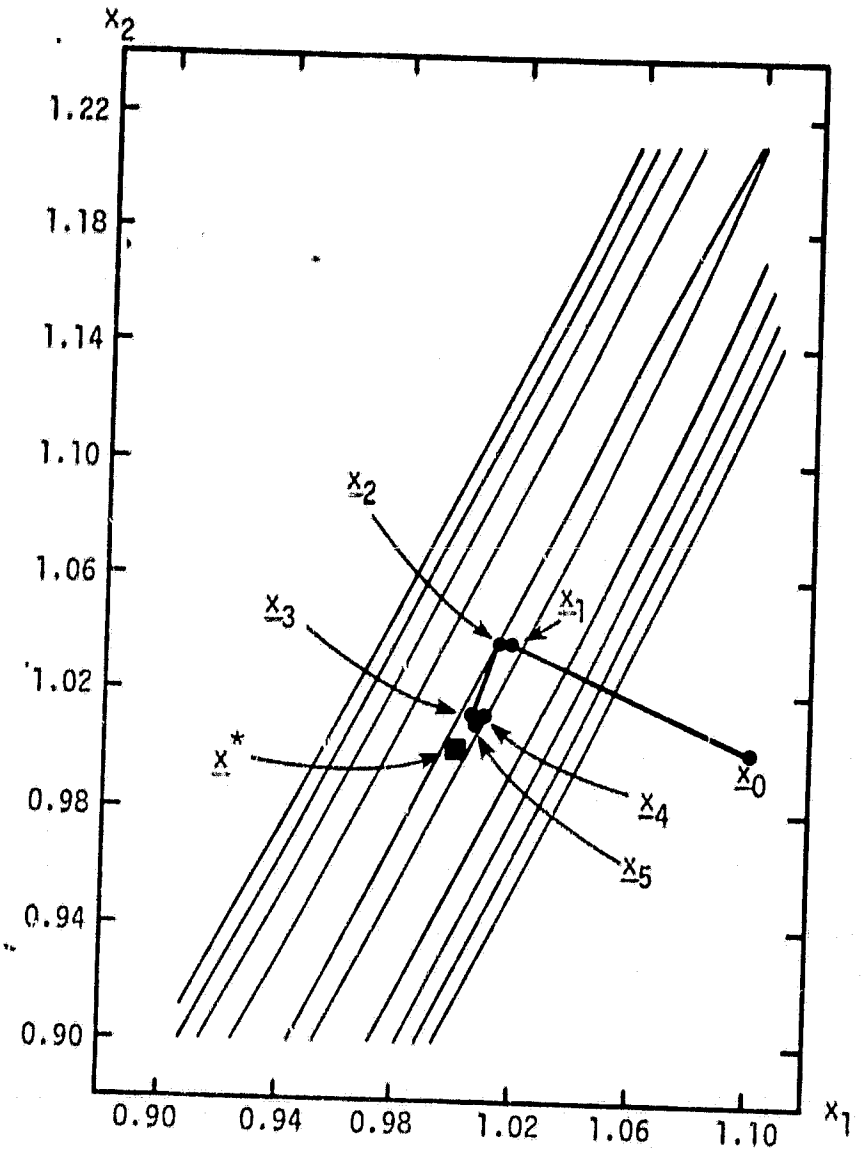


Figure 5.4 Modified D-F-P Example

For simple functions, different starting points may be used to find all of the local minima and hence, the global one. In Chapter VII, it will be seen that the cost function employed in this work has far too many local minima for this technique to be practical. A scheme for examining ranges of the variables will be shown to be applicable.

CHAPTER VI

THE COST FUNCTION

With analytic expressions for the noise annoyance (Chapter III) and the system constraints (Chapter IV), it is possible to formulate a mathematical criterion, the cost function, by which the amount of any change may be judged. As explained in the preceeding chapter, including the constraints as penalties in the cost function allows the use of standard, unconstrained optimization techniques, which are more practical for this work than are the so-called primal methods. While the annoyance is the primary quantity to be minimized (with the penalties as a necessary adjunct of the cost), it may be desired to include other properties of the system in the cost, such as the fuel and time of flight required for each trajectory.

The Penalty Functions

All of the constraints developed in Chapter IV, Eqns. (4-9) and (4-13) through (4-20) are in the form

$$h_i(\underline{x}) \leq c_i \tag{6-1}$$

as required by Eqn. (5-1), even though the unknown $\underline{x}$ does not appear explicitly in any of them. Penalties in the form of Eqn. (5-3) may then be written for each constraint as follows:

[Dynamic and Passenger Comfort Constraint, Eqns. (4-9), (4-13)]

$$P_1 = \sum_{\text{traj.}} \sum_{\substack{\text{corner} \\ \text{points}}} \left(\max\left[0, \max\left\{ \frac{\bar{V}^2}{C_2 g} , \frac{\bar{V}}{(c_1+c_2c_3)\min(\delta r_1 \delta r_2 \delta r_3)} \right\} - R\right]\right)^2 \tag{6-2}$$

$$P_2 = \sum_{\text{traj.}} \sum_{\text{segments}} (\max[0, \tan\gamma_{dmax} - \frac{dz}{dl}, \frac{dz}{dl} - \tan\gamma_{cmax}])^2 \quad (6\text{-}3)$$

[Geometric Feasibility Constraints, Eqns. (4-14), (4-15)]

$$P_3 = \sum_{\text{traj.}} \sum_{\substack{\text{corner} \\ \text{points}}} (\max[0, \left[\frac{R_{min}}{\tan\frac{\alpha}{2}}\right] + \left[\frac{R_{min}}{\tan\frac{\beta}{2}}\right] - l_2])^2 \quad (6\text{-}4)$$

$$P_4 = \sum_{\text{traj.}} (\max[0, \frac{R_{min}}{\tan\frac{\alpha}{2}} - l_1])^2 \quad (6\text{-}5)$$

[Separation and Crossover Constraints, Eqn. (4-16), (4-17)]

$$P_5 = \sum_{\substack{\text{all traj.} \\ \text{pairs}}} \sum_{i=1}^{s} (\max[0, d_{min} - \hat{y}_i])^2 \quad (6\text{-}6)$$

$$P_6 = \sum_{\substack{\text{all traj.} \\ \text{pairs}}} \sum_{i=1}^{s-1} (\max[0, -\hat{y}_i\hat{y}_{i+1}])^2 \quad (6\text{-}7)$$

[Final Heading Constraint, Eqns. (4-18), (4-19)]

$$P_7 = \sum_{\text{traj.}} (\max[0, 135° - \delta])^2 \quad (6\text{-}8)$$

$$P_8 = \sum_{\text{traj.}} (\max[0, l_1 - \tfrac{1}{2}l])^2 \quad (6\text{-}9)$$

[Threshold Constraint, Eqn. (4-20)]

$$P_9 = \sum_{j\in J} (\max[0, \{\sum_{i=N^*}^{N_f} \max(0, L_{A,max,i} - L_{A,Thresh})\}])^2 \quad (6\text{-}10)$$

The Total Cost

Adjoining the penalties to the desired annoyance measures, say NII, gives the total cost function:

$$f(\underline{x}) = w_A A(\underline{x}) + \underline{\mu}^T \underline{P}(\underline{x}) \qquad (6\text{-}11)$$

where $A(\underline{x})$ is the annoyance (NII), w_A is the annoyance weight, $\underline{\mu}$ is the m-vector of penalty weights, and $\underline{P}(\underline{x})$ is the m-vector of penalties (there are m constraints). As before, the variable $\underline{x}$ is present only in a formal sense; it does not appear explicitly in these terms in this work.

They annoyance weight w_A governs the relative importance of the annoyance, keeping it from being dominated by the penalty terms. This is most important in cases where the optimal solution is on a constraint boundary. In such a region, $\underline{\mu}^T \underline{P}(\underline{x})$ will in general be nonzero, but as discussed in Chapter V, small values are acceptable. The purpose of w_A then is to insure that whatever reduction can be made in the total cost corresponds to a decrease in the annoyance, rather than a decrease in penalties that represent tolerable constraint violations. When the constraints are violated excessively, the attendant large penalties naturally should comprise the dominant portion of the cost.

Experimentally, the following values of w_A and the μ_i have been found to give satisfactory results when NII is used for the annoyance measure:

$\mu_1 = 10^4 - 10^6$	(lateral dynamic, passenger comfort)
$\mu_2 = 10^2 - 10^6$	(longitudinal dynamic)
$\mu_3 = 10^4 - 10^6$	(geometric feasibility)
$\mu_4 = 10^4 - 10^6$	(geometric feasibility, first segment)

$\mu_5 = 10^3$ (separation)

$\mu_6 = 10^3$ (crossover)

$\mu_7 = 10^4 - 10^6$ (final heading)

$\mu_8 = 10^4 - 10^6$ (final heading)

$\mu_9 = 10^3$ (threshold noise).

The rather wide range in μ_2 merits some explanation. It has been often found[1] in this work that the optimum trajectories will have the maximum vertical slopes allowed by the longitudinal dynamic constraint, Eqn. (4-9). This is intuitively expected, since the maximum slopes give the highest average altitude, the largest average distances from all the populace, and hence, the lowest average noise levels. Because of this tendency in the solutions, the associated penalty (P_2) generally is non-zero. Starting with a small value of μ_2 allows the optimization to work on the ground tracks without hinderance from penalties on the profiles (vertical components). After several iterations, μ_2 may be increased if excessive longitudinal constraint violations are occurring.

Modifying the Cost Function: Inclusion of Fuel and Time of Flight

An example of altering the cost function to include other attributes is a study conducted to examine the relationship between the noise

---

1 In cases where a trajectory segment passes over a region of low population density, increasing its slope does not significantly lower the noise levels and hence, the total community annoyance. The optimization stops when the change in cost between successive iterations is less than some criterion. Therefore such a segment may not be exactly optimum, but practically it is so. Fuel consumption rates that vary with climb and descent rates are not included in this analysis.

annoyance, and the fuel and time of flight needed. It is assumed that the latter two vary linearly with the flight path length only. That is, constant power settings and noise propogation characteristics are used. The resulting term to be added to the cost is

$$f_{F,TOF}(\underline{x}) = \sum_{traj.} W_{F,TOF} l_T \tag{6-12}$$

where $W_{F,TOF}$ is the fuel and time of flight weight, and l_T is the trajectory length. Eqn. (6-11) becomes

$$f_1(\underline{x}) = w_A A(\underline{x}) + f_{F,TOF}(\underline{x}) + \underline{\mu}^T \underline{P}(\underline{x}). \tag{6-13}$$

Using $f_1(\underline{x})$ as the cost function, a relation between the annoyance and $f_{F,TOF}$ was obtained by varying $w_{F,TOF}$. This investigation is reported in Chapter VIII.

CHAPTER VII

INTEGRATION OF THE MODELS

The annoyance, flight path, and cost function models are combined with the chosen optimization method to create a system for analyzing the aircraft noise annoyance problem and the feasibility of remedying it through trajectory modifications. While the components of the system are essentially modular (and hence, capable of being altered individually without replacing the entire system), some linkage among them is required. Further, some special considerations must be given to larger airports and to some population distributions.

The Flyover/Noise Simulation: Small Airports

At any airport, there are, in general, a number of flight paths (both for takeoffs and landings) in use on a given day, with different types and numbers of aircraft using these trajectories. In this work, it is assumed that for the period of time under consideration, the type of each path (takeoff or landing) and the runway to which it is assigned do not change. In addition, the noise level data used is that for landing aircraft with constant power and control settings.[1]

Calculation of the annoyance measure requires information about the distribution of the sound levels (L_{dn}) on the ground; for the current population distribution model, only the sound levels at the cell

---

[1] These assumptions are not fundamental to the operation of the system. A more sophisticated noise model, for instance, that employs thrust dependent noise data could be used instead of the present one.

centroids are needed. First, the position of each type of aircraft, on each trajectory, is sampled at equal time intervals (10 sec.). Then Eqn. (3-1), along with data from Table 3.1, is used to compute the sound level (L_A) at each centroid due to each type of aircraft, from every sample position and trajectory. This computational approach is practical so long as the number of flight paths and aircraft types is small. Referring to Appendix A, these levels for all of the aircraft are added on an energy basis,[2] and the average day-night level (L_{dn}) at each centroid may then be calculated from Eqn. (3-5).

Knowledge of the L_{dn} distribution, along with the population distribution, allows the calculation of either LWP (c.f. Eqn. (3-8)) or NII (c.f. Eqn. (3-9)); however, evaluation of HAPN is not necessarily as straightforward. This measure requires that the sound level (L_A) be calculated at each centroid, at each sampling instant, and that the maximum values (day and night) be retained. At smaller airports, where the flight operations are dispersed and non-overlapping in time, the L_A values depend upon only one aircraft at any sampling instant. For a larger number of operations where the overlap (in time) is unavoidable, the sequence of flight operations must be explicitly included in the L_A calculations, since at every sampling instant the types of

---

[2] It is not necessary to compute L_A (at each centroid) for every sampling instant, since these levels will themselves be averaged (using energy addition) to get L_{dn}. Only the energy term, Eqn. (A-4), must be computed and accumulated (correctly weighted, c.f. Eqn. (3-5)) into the total centroid energy for every sampling instant. If n aircraft of the same type fly on a give trajectory, then their total energy contribution at any centroid is n times the energy added by one aircraft.

aircraft flying will determine the value of L_A at any centroid. In this study, only a situation of the former type has been considered for the purpose of comparing HAPN with LWP and NII.

Large Airports and the Equivalent Aircraft Concept

Summing (on an energy basis) over all trajectories, sample points, and aircraft types may become impractical as the number of any of these grows, depending upon the computing resources available. At large airports, such as the Lambert-St. Louis facility in St. Louis, Missouri, the number of trajectories and aircraft (fifteen paths and five hundred operations per day) precludes the straightforward approach of the last section. In such a case, the summation over aircraft types may be removed by replacing the mixture of planes on each trajectory with a single equivalent aircraft, whose sound level distribution is the same as if all the aircraft assigned to that path were flying together on it. For this treatment to be valid, it is necessary that either NII or LWP be used as the annoyance measure. The reason is that both of them are functions of the L_{dn} distribution, which depends only on the total sound energy distribution, integrated over time (one 24-hour period) and not on the time history of the energy contributions (except for the distinction between day and night occurrences).

The exact expression for the total sound level (using any power intensity related scale) contributed from n sources is (see Appendix A)

$$L_{TOT} = 10 \cdot \log_{10} \sum_{i=1}^{n} 10^{L_i/10} \qquad \text{(A-6)}$$

with, from Eqn. (3-1)

$$L_i = c_{1,i} - c_{2,i} \cdot \log_{10} r, \tag{7-1}$$

where the subscript i is the aircraft index and n is the total number of aircraft. A simple, approximate expression for Eqn. (A-6) may be obtained in the following way. For all the aircraft on a given trajectory, the appropriate L_i's are added (on an energy basis, and using the L_{dn} convention which weights nighttime energy terms with a factor of ten), i.e. an equivalent level L^* may be written

$$L_k^*(r) = 10 \cdot \log_{10} \sum_{i=1}^{n} w_i 10^{L_i(r)/10} \tag{7-2}$$

where the subscript k denotes that this is the L* contribution from the k-th trajectory, and w_i is the same time-of-day factor as in Eqn. (3-5). If $L_k^*(r)$ vs. r is plotted for a variety of values of r, an excellent approximation of the resulting data is the expression

$$L_k^* \simeq c_{1,k,eq} - c_{2,k,eq} \cdot \log_{10}(r) \ , \tag{7-3}$$

where $c_{1,k,eq}$ and $c_{2,k,eq}$ are the equivalent aircraft noise coefficients for the k-th trajectory. An example of this is shown in Figure 7.1. The largest value of r was chosen to be 60,000 meters, since that is the largest value of the range that would be encountered in the near zone area (32,000 meter radius) of an airport. In this example, the estimated correlation coefficient squared in the least square-error fit (dotted line) was $r^2 = 0.9995$. (Spacing the abscissa values such that $r_{j+1} = ar_j$, where a is a constant, results in their being equally spaced on a logarithmic scale, and gives a better curve fit.)

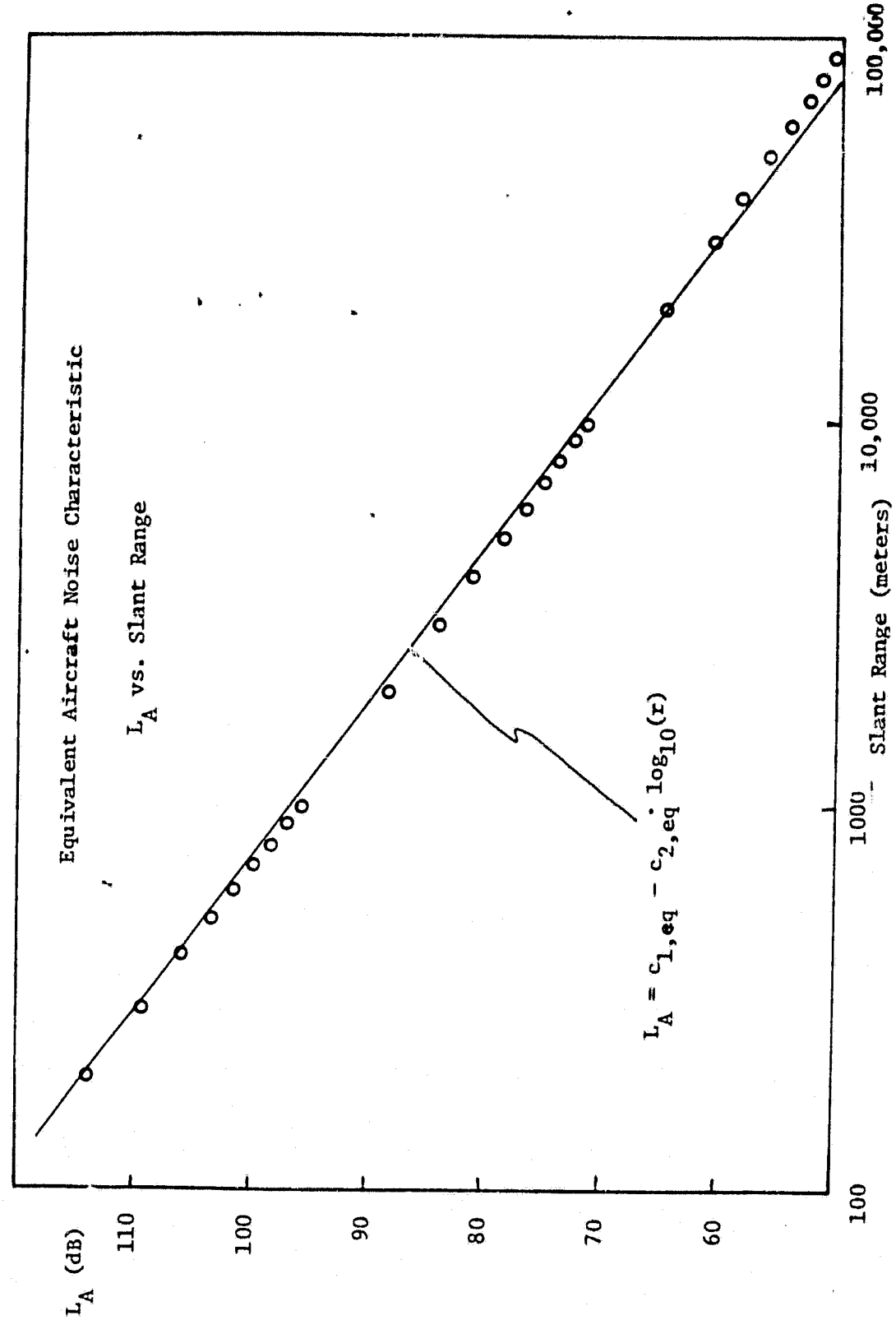


Figure 7.1 Equivalent Aircraft Noise Data Fit

With the coefficients $c_{1,k,eq}$ and $c_{2,k,eq}$ for all the flight paths, the total L* at any point on the ground, at each sampling instant is just the sum (on an energy basis):

$$L_t^* = 10 \cdot \log_{10} \sum_{k=1}^{N_T} 10^{L_{k,t}^*/10} \quad , \tag{7-4}$$

where N_T is the number of trajectories, and t is the sampling index. Since the time-of-day weighting has already been included in L_t^*, the expression for L_{dn} becomes (c.f. En. (3-5))

$$L_{dn} = 10 \cdot \log_{10} \frac{\sum_{t=1}^{N} 10^{L_t^*/10}}{N} \quad , \tag{7-5}$$

where N is the number of noise level samples.

This method of using equivalent aircraft coefficients reduces the number of calculations needed to determine L_{dn} at each grid cell by a factor approximately equal to the number of aircraft types that are operating in an area. Since at many airports this number is frequently on the order of ten, a considerable savings in computation time may be achieved. (Minimal computation time is required to determine the equivalent aircraft noise coefficients.)

As it is the method relies upon the assumption that $c_{1,i}$ and $c_{2,i}$ are constant for each aircraft type. If varying thrust levels and control settings are to be considered, it might be possible to make these coefficients functions of time, and for a given airport, predict their values (and hence, those of the equivalent aircraft coefficients) by estimating the time histories of the thrust and controls. A second disadvantage of the method is that L_A values are not computed at each

sampling instant. Thus, the threshold noise constraint either must be omitted or altered so that the limitation is in terms of L_{dn} in each grid cell, rather than the maximum A-level experienced. Finally, the errors introduced in the L_{dn} values by the equivalent aircraft method are small ($\leq$ 1 dB); however, the nonlinear intensity weighting function, $W(L_{dn})$, may amplify these errors, giving NII and LWP values that differ by several percent from those calculated exactly. The amount of error in NII and LWP changes with the sets of trajectories, but the locations of the local minima do not differ significantly in comparisons of the exact and approximate calculations.[1]

Equivalent Aircraft Coefficients and the Assignment Problem

An interesting feature of the equivalent aircraft method is that the noise level coefficient $c_{1,k,eq}$ (k is the trajectory index) has a strong dependence only on the total number of aircraft on a particular flight path, as long as there is a uniform mixture of aircraft types present. A plot of $c_{1,k,eq}$ vs. the L_{dn}-equivalent number[2] of aircraft per trajectory (for the Lambert/St. Louis International Airport) is shown in Figure 7.2. The dotted line is the least square-error fit and has the form

---

[1] For example, using the Phoenix Sky Harbor Airport case, the error in NII for the nominal (currently specified) trajectories is 2.1% and for the optimum set, 0.2%. The corner pts. (equivalent aircraft method) differ from the optimum ones (exact calculations) by no more than 100 meters.

[2] The L_{dn} convention of weighting a nighttime sound level with a factor ten is equivalent to adding the sound levels, on an energy basis, of ten identical sound level occurences. Hence, one nighttime flight is equivalent (in L_{dn} calculations) to ten daytime flights.

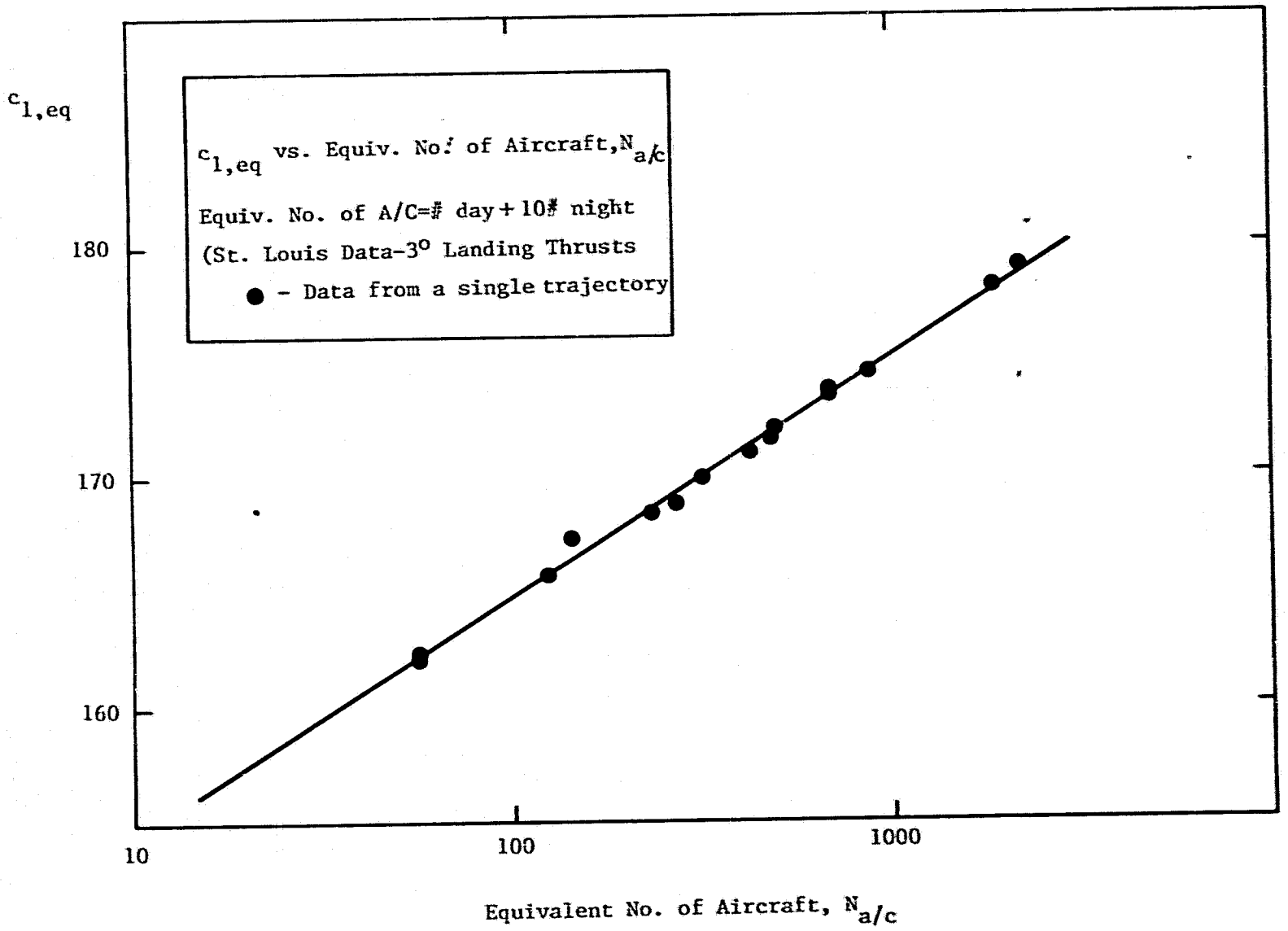


Figure 7.2 Plot of $c_{l,eq}$ vs. Equivalent Number of Aircraft

$$c_{1,k,eq} = a + b \cdot \log_{10} N_{a/c} \,, \tag{7-6}$$

where $N_{a/c}$ is the L_{dn}-equivalent number of aircraft per trajectory and in this case, a = 144.65 and b = 10.16. This fit has an estimated correlation coefficient squared, $r^2 = 0.993$. As for the coefficient $c_{2,k,eq}$, the data (also for the Lambert-St. Louis case) are not as strongly correlated with $N_{a/c}$. Figure 7.3 shows a log-log plot of $c_{2,k,eq}$ vs. $N_{a/c}$, with the least square-error fit (dotted line) for $N_{a/c} \geq 250$. The functional form is

$$c_{2,k,eq} = aN_{a/c}^{b} \,, \tag{7-7}$$

where a = 23.44 and b = 7.89×10^{-3}; the estimated correlation coefficient squared is $r^2 = 0.894$. Restricting $N_{a/c}$ to values greater than 250 allows for a good fit of the data with a simple functional form. For $N_{a/c} < 250$, Eqn. (7-7) will not accurately predict $c_{2,k,eq}$, and an individual fit of the data, as in Figure 7.1, will be necessary.

These relations between the coefficients and $N_{a/c}$, Eqns. (7-6) and (7-7), have a bearing on the problem of how to assign aircraft to the different trajectories, with the community annoyance from noise as the criterion.

In this work, the assignment problem is not considered (the assignments are assumed to be known and remain constant for a given airport); however, it will be a necessary addition in subsequent refinements of this study. The aforementioned relations may be useful in formulating a performance criterion to be used in the initial assignment of aircraft, or in altering an existing plan.

C - 2

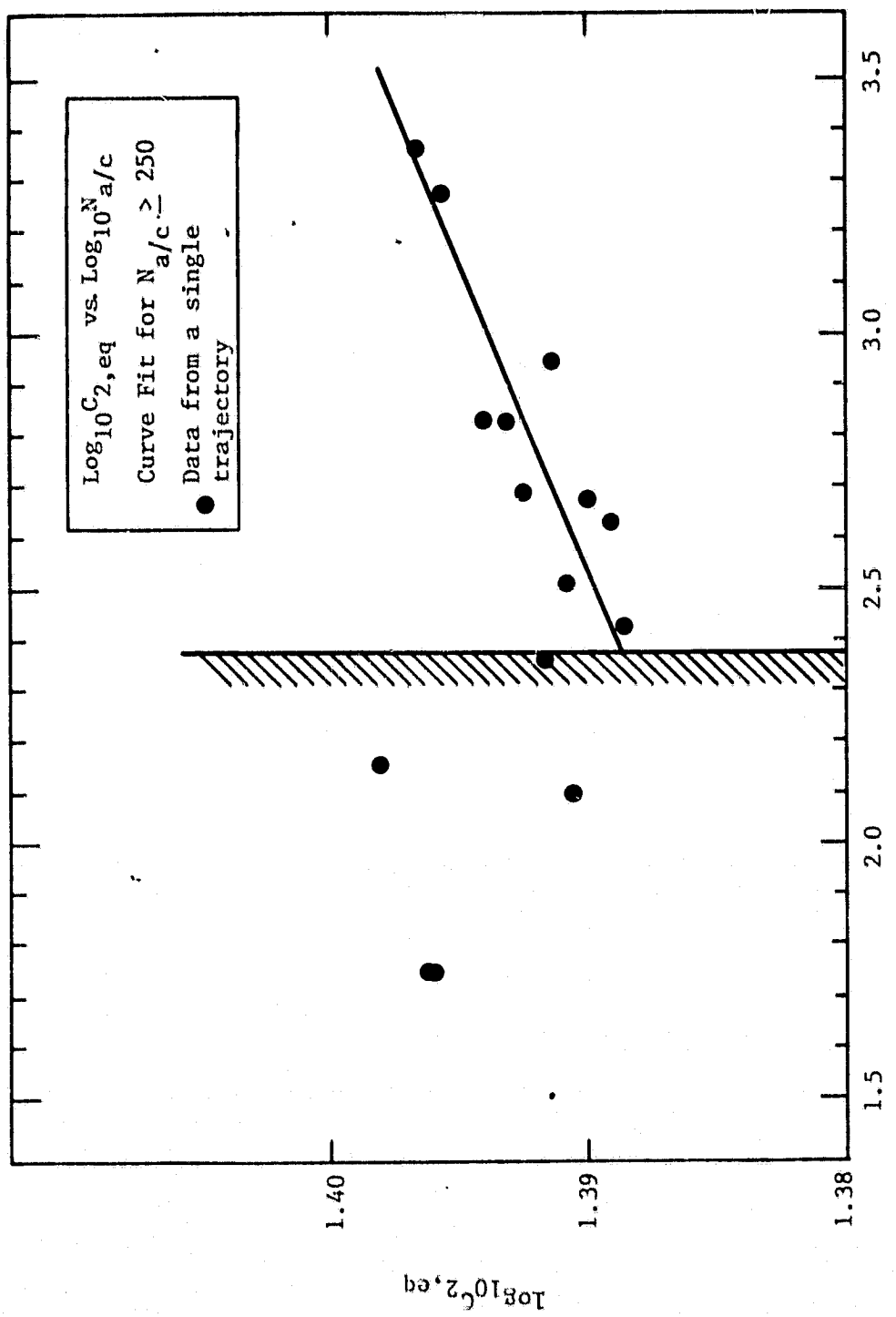


Figure 7.3 Plot of $Log_{10}C_{2,eq}$ vs. $Log_{10}N_{a/c}$

The Integrated System

A general summary of the systems operation is provided by the flowchart in Figure 7.4.

The input required is: 1) the population distribution data, 2) the number of each aircraft type and their time-of-day designations (day or night), 3) the noise level characteristics (either by individual aircraft type or equivalent aircraft noise coefficients), 4) the constraint data (minimum turning radius, threshold noise level, etc., and penalty weights), 5) an initial set of flight paths (specified by their end points and corner points), and 6) the stopping criteria (the maximum number of iterations, and the lower limit on relative changes in the cost function).

For the initial trajectories, the total cost must be evaluated, before starting the iterative search. (If the system is being used only for evaluation of the trajectories, then no search is performed, and the process stops after this step.) First, the annoyance measure (e.g., NII, LWP, HAPN) is evaluated. Then the constraints are tested: a violated constraint results in a positive penalty; otherwise, the penalty is zero. Finally, the total cost is computed by adding the penalties to the annoyance. As shown in Chapter VI, other cost terms may be added to the total; the results of one such modification appear in Chapter VIII.

Now the optimization loop begins. After each search step, a test is made to determine if any of the stopping criterion is satisfied. If not, the iteration continues; if so, the process stops and the "minimum" annoyance trajectories are determined. In fact, the set determined may

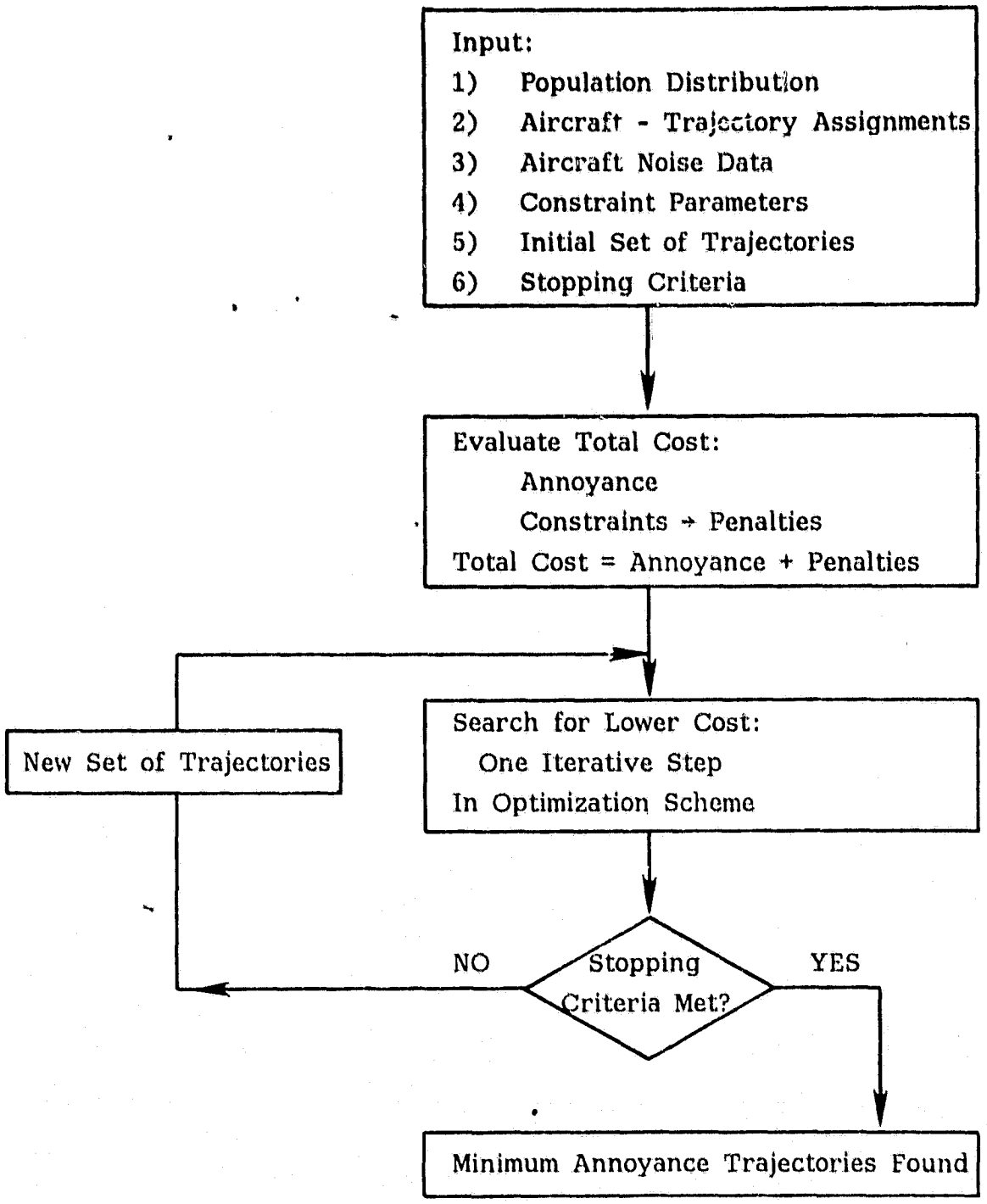


Figure 7.4 System Flowchart

be only a locally minimum set (a problem discussed in the next section), or perhaps not even that (the maximum limit on the number iterations allowed may have been reached); however, because of the type of optimization method used (Chapter V), each successive set of paths will result in a lower total cost (and a lower annoyance, if the penalties are zero).

Besides a limit on the number of iterations, the stopping criteria include the tests: 1) Is the gradient of the cost function zero? (i.e., the search has found a local minimum), and 2) Is the relative difference between successive values of the cost negligible? (i.e., the search has come close enough[3] to a local minimum to warrant no further searching).

Local Minima and the Population Distribution

A problem with the optimization, already expressed in Chapter V, is that of local minima in the cost function. As an example of the problem, consider a cost function of two variables, for which the contours are plotted in Figure 7.5. Points A, B, and C are the local minima; point B corresponds to the lowest value of the cost, and hence, is the global minimum. The difficulty associated with this situation is that of finding point B by use of any of the optimization methods discussed in Chatper V. Given an initial point from which to search, each

---

[3] The error in the solution, $E = \underline{x}^k - \underline{x}^*$, where $\underline{x}^k$ is the k-th iterative solution and $\underline{x}^*$ is the exact solution, may in fact be large. If g_k is small, though, and the relative difference between successive costs is small, then further searching may not lead to a significant reduction in the cost. Lacking other information, it may be assumed that, practically, a local minimum has been located.

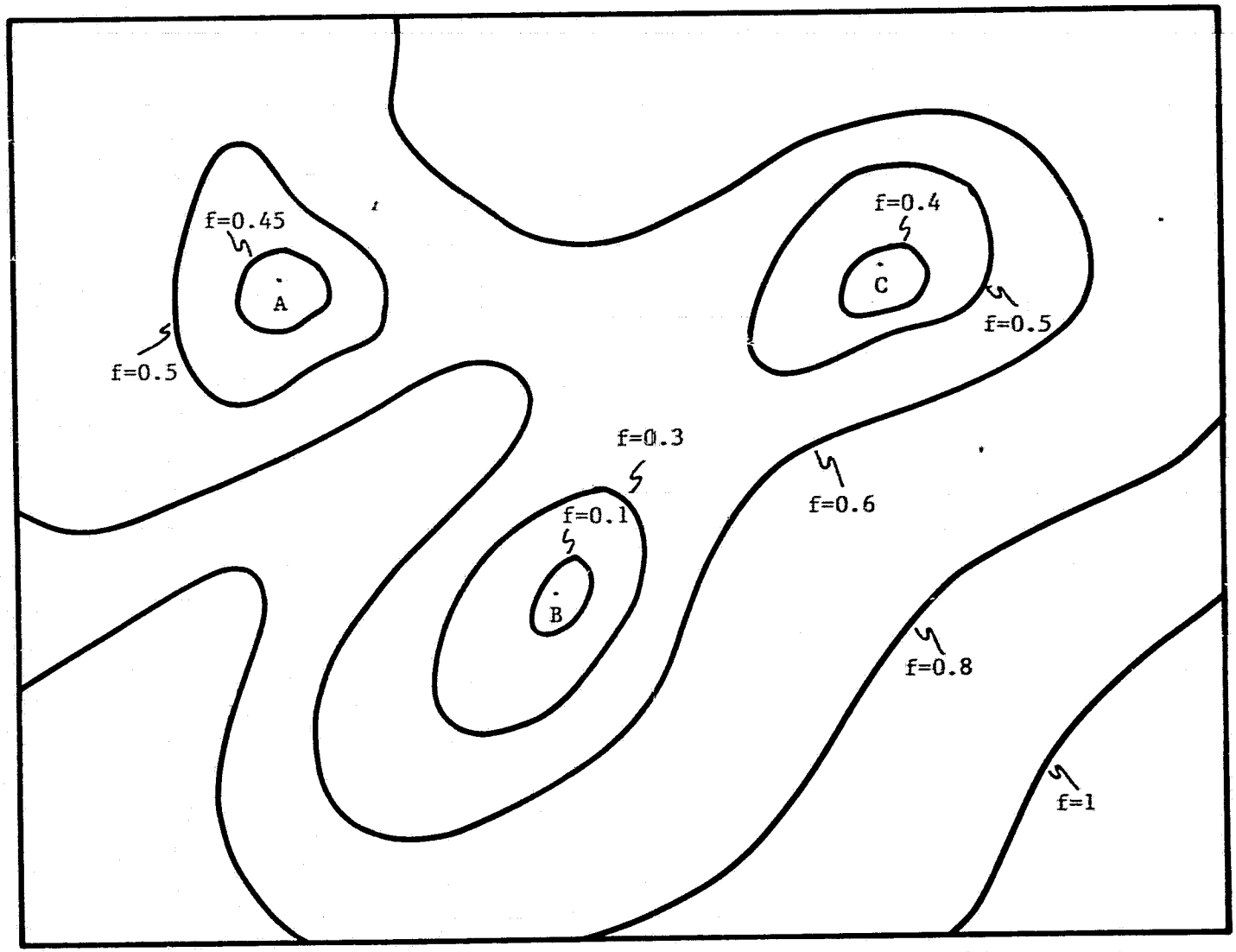


Figure 7.5 Several Local Minima

of those methods will converge to one of the local minima, but will not automatically find the global minimum. Which local minimum is found depends upon the starting point in the search. The conventional technique used to find the global minimum is to perform the optimization a number of times, from different initial points. In cases where the cost function has only a few minima, this approach may indeed allow all of them to be found. If the total number of local minima is unknown, but judged to be small, it is often possible to estimate the probability of overlooking a number of minima. That probability may then be made acceptably small by repeating the optimization with a large enough number of starting points. Ref. [35] and [36] are good examples of this approach.

The cost function used in this work has an extremely large number of local minima. This is demonstrated by considering the population distribution shown in Figure 7.6a along with the simple straight-path trajectory. This path is at a constant altitude (300 m.). There are 1000 people at each dot on the map. In Figure 7.6b, the resulting NII values versus θ are plotted (a typical mixture of commercial aircraft is assumed for generating the noise levels).

Even though NII is computed from noise levels experienced by the entire community, the local minima exhibited in Figure 7.6b demonstrate the influence of high levels that occur locally within the community. For instance, at $\theta_1 = 20°$, the trajectory passes close to one population center and almost directly over another; the resulting noise level at these points, and hence, their contributions to the NII will be high: a peak occurs in Figure 7.6b at $\theta_1 = 20°$. Conversely, at $\theta_2 = 26°$, the

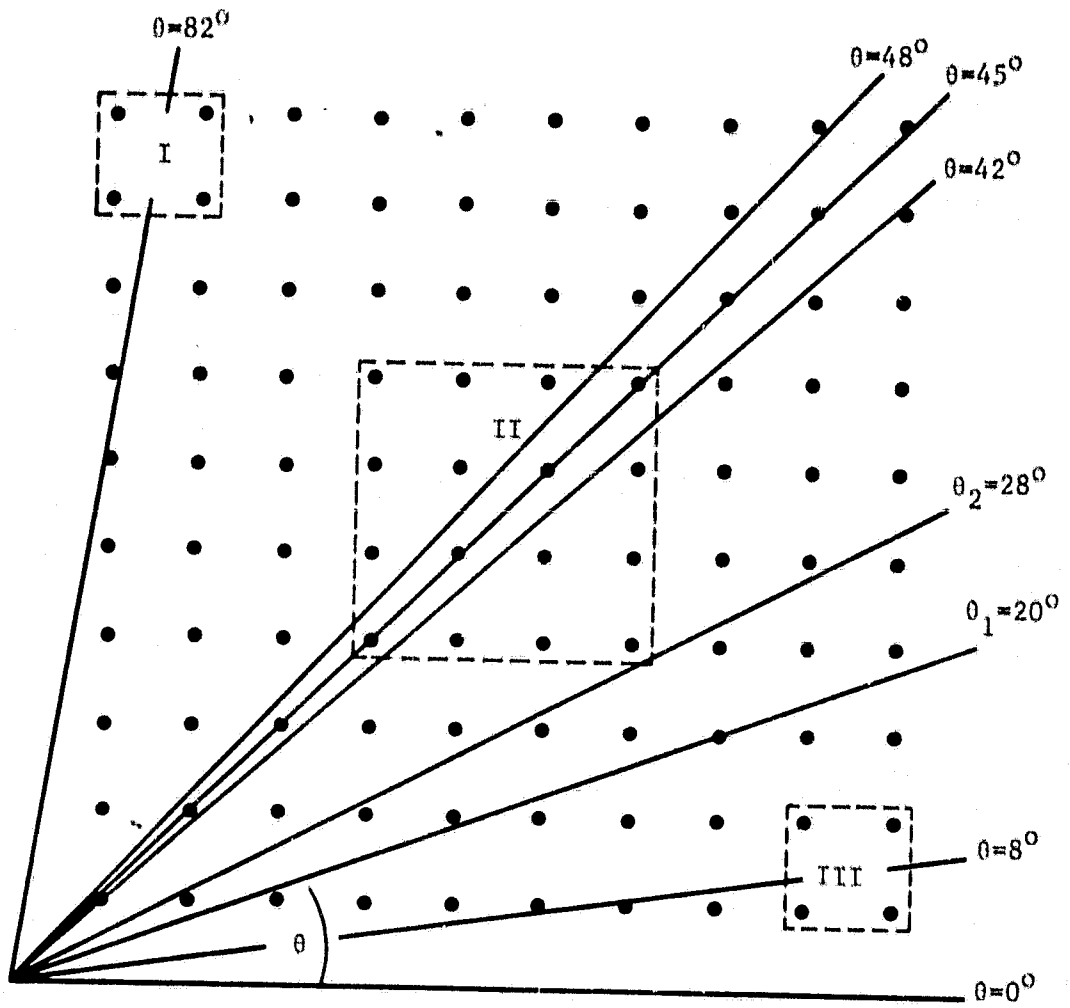


Figure 7.6a Uniform Discrete Population Distribution and Straight Flight Paths

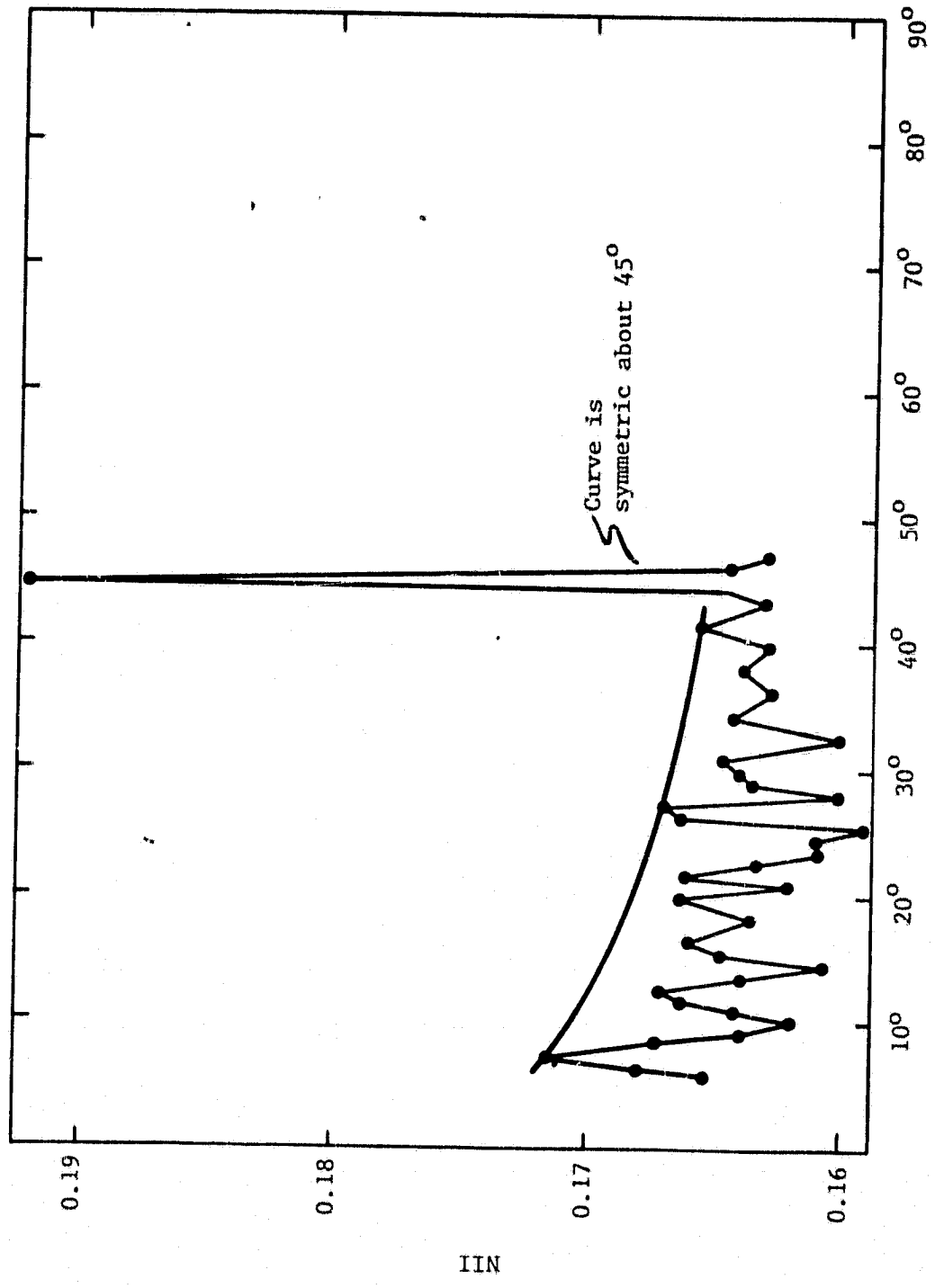


Figure 7.6b NII vs. Trajectory Direction

population points nearest to the trajectory are farther away from it than those in the first case. The noise levels they experience are lower, and a minimum in the NII occurs at $\theta_2 = 26°$. (At $\theta = 45°$, the sharp peak is a local effect caused by the large number of population centers directly beneath the path.)

This example demonstrates two important problems. First, there are many local minima in the cost function, caused mostly by the use of a particular representation of the population distribution.[4] In a realistic case, there are more population centers and trajectories, which lead to an even larger number of local minima. Second, the arbitrary lower limit on L_{dn} values to consider (55 dB) for use in the NII has altered the results that would be expected intuitively. The envelope of NII peaks in Figure 7.6b is seen to decrease as θ increases, i.e., as more of the population receives higher noise intensities, the community annoyance decreases. This occurs because the denominator in Eqn. (3-9) increases, in this case, faster than LWP. Once again, the inappropriateness of NII for use in comparing the annoyance from different trajectories in a given community is in evidence.

A dramatic change in the number of local minima in this example occurs if NII is calculated without a lower limit on the L_{dn} interval (c.f. Eqn. (3-9)) but with the total population fixed. This modified version of NII will be denoted NII′. Figure 7.6c shows the results

---

[4] In each population grid cell, the people are assumed to be concentrated at the geometric centroid of the cell. The overall result is a discontinuous (discrete) density distribution.

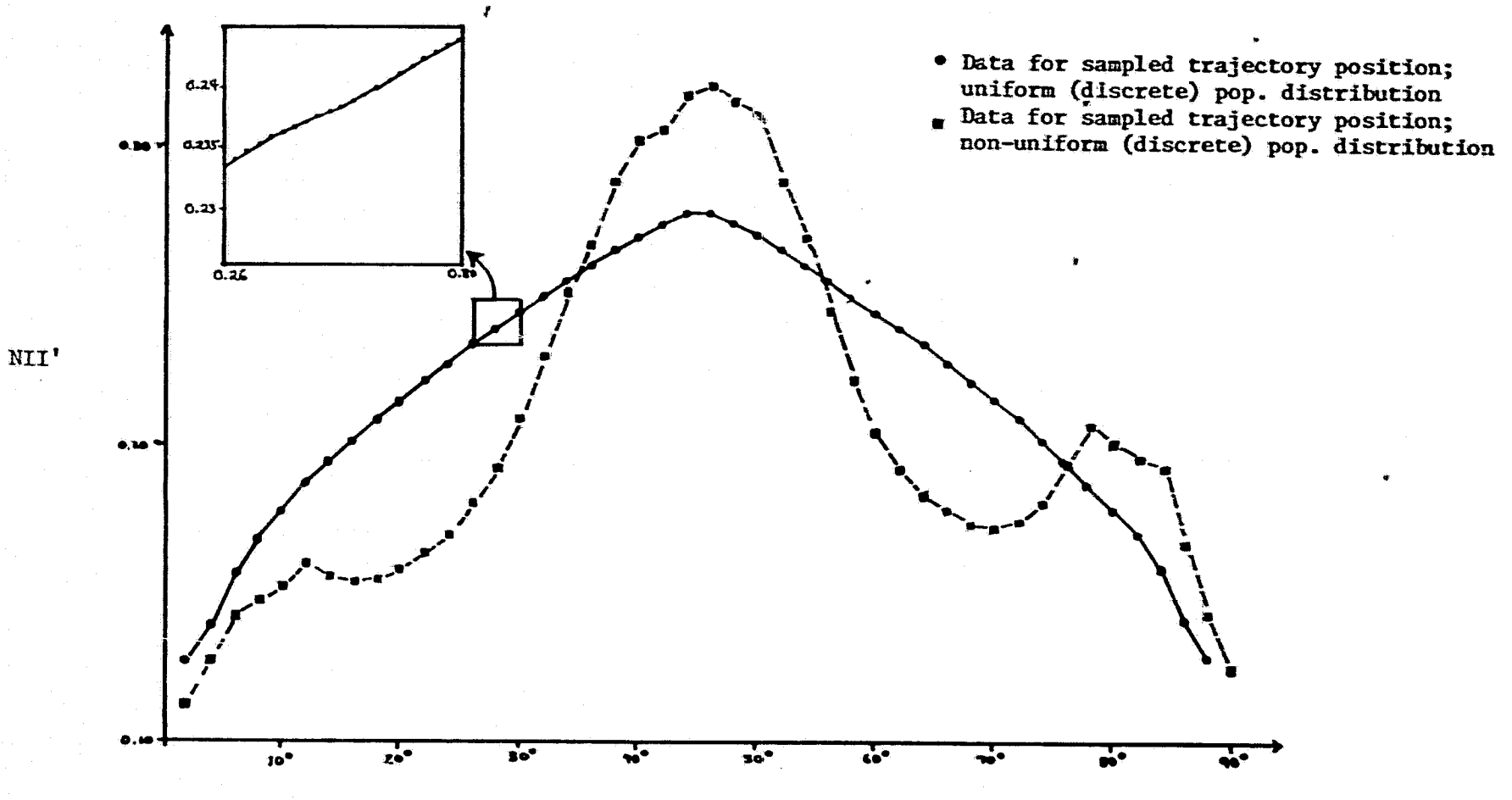


Figure 7.6c NII´ vs. Trajectory Direction

(solid line). The complete disappearance of all the local minima is fortuitous in this case (the influence of local effects still persists, though, as seen in the inset); it occurs as a consequence of the uniform (discrete) distribution of people and also because of the particular degree of freedom given to trajectory variations here.[5]

There remains the dilemma of local minima, caused by macroscopic variations in the population density. Again, referring to Figure 7.6a, assume that in Region I, there are 20,000 people at each population center; in Region II, 25,000 people at each center; in Region III, 10,000 at each center; all other centers have 1,000 people. The resulting values of NII′ vs. θ appear in Figure 7.6c (dotted line). At $\theta = 16°$ and $\theta = 70°$, there are local minima, caused by the lack of large numbers of people close to these paths. Because the truly large clusters of people are in Regions I, II and III, it is reasonable to conclude that these minima are real effects of the population distribution, and not artifices of its approximate representation. Note that at $\theta = 8°$, 82°, 42°, 48°, there are slight dips in NII′, caused by the flight path's avoidance of all the nearest population centers.

The possibility of artificial minima remains, though. Note that at $\theta = 8°$, 42°, 48° and 82°, there are inflections in NII′. Although not minimum points, there correspond closely to minima in Figure 7.6b. Further, no such behavior is seen in Figure 7.6c (solid line) for the

---

[5] Only θ may vary, which limits the possible paths. In the next section, it will be seen that trajectories with several segments still have local minimum problems with this type of population distribution model.

uniform distribution. It may be suspected then that artificially produced minima may occur even with the use of NII', depending upon the actual distribution of people and the use of this population distribution model.

Global Searching

Regardless of the cause of the local minima, some procedure for determining the optimum solution must be developed. First, it must be recognized that finding the exact optimal set of flight paths is not as important as determining a set (if one exists) that produces a significant reduction in the annoyance measure, compared with the set presently used (nominal set). A given set will be considered optimum if there is not another set, significantly different, with a lower cost function value.

The definition of "significantly different" is developed in terms of the statistical lateral dispersion of individual flights from the specified trajectories. Galloway [37] states that the lateral dispersion of landing aircraft is approximately 750 m. at a distance of 16,000 m. from the airport. This corresponds to an angular dispersion of approximately 3°. For takeoffs, where the paths are less tightly controlled, the lateral dispersion is given in terms of the standard deviation (σ) of the side-line distance (y) from the specified path. For straight-out takeoffs, the standard deviation grows linearly with the distance l and is given by

$$\sigma_y = 0.08\ l\ . \tag{7-8}$$

After a turn, the dispersion is expressed as

$$\sigma_y = 3.5 \times 10^3\ \sigma_z/\bar{z}\ , \tag{7-9}$$

where $\sigma_z/\bar{z}$ (the standard deviation in altitude dispersion divided by the average altitude during the turn) is generally a constant; the value $\sigma_z/\bar{z} \simeq 0.2$, for variable weight transports, will be used here.

Eqn. (7-8) may be used to estimate the angular dispersion on straight segments of a takeoff by assuming that all of the flights will lie within $2\sigma_y$ of the specified paths (a Gaussian distribution would contain 95% of the paths within $2\sigma_y$ of the mean path). The outermost path, to either side of a straight segment, will then form an angle, θ_{disp}, with, the segment, given by

$$\theta_{disp} = \tan^{-1} \frac{0.16\ 1}{1} = 9^{\circ}. \qquad (7\text{-}10)$$

A set of trajectories will be considered "significantly different" from another set only if any path lies outside the dispersion limits of the corresponding path in the other set. As an example of how this may be used to restrict the amount of searching required in the optimization process, the flight paths at the Lambert-St. Louis International Airport will be considered. Figure 7.7a shows the twelve nominal takeoff paths and three nominal landing paths. The shaded regions represent the predicted dispersion about each path. Note that the trajectories with turns possess more average dispersion, c.f. Eqn. (7-9); this is due to the extra drifting during a turn.

Searching for the optimum trajectory set within the dispersion limits of this set is unprofitable since such an optimum would not by definition be significantly different from the nominal set. Of course, it may happen that the nominal set <u>is</u> the optimum, but regardless, any searching must occur "outside" the disperson boundaries. Different

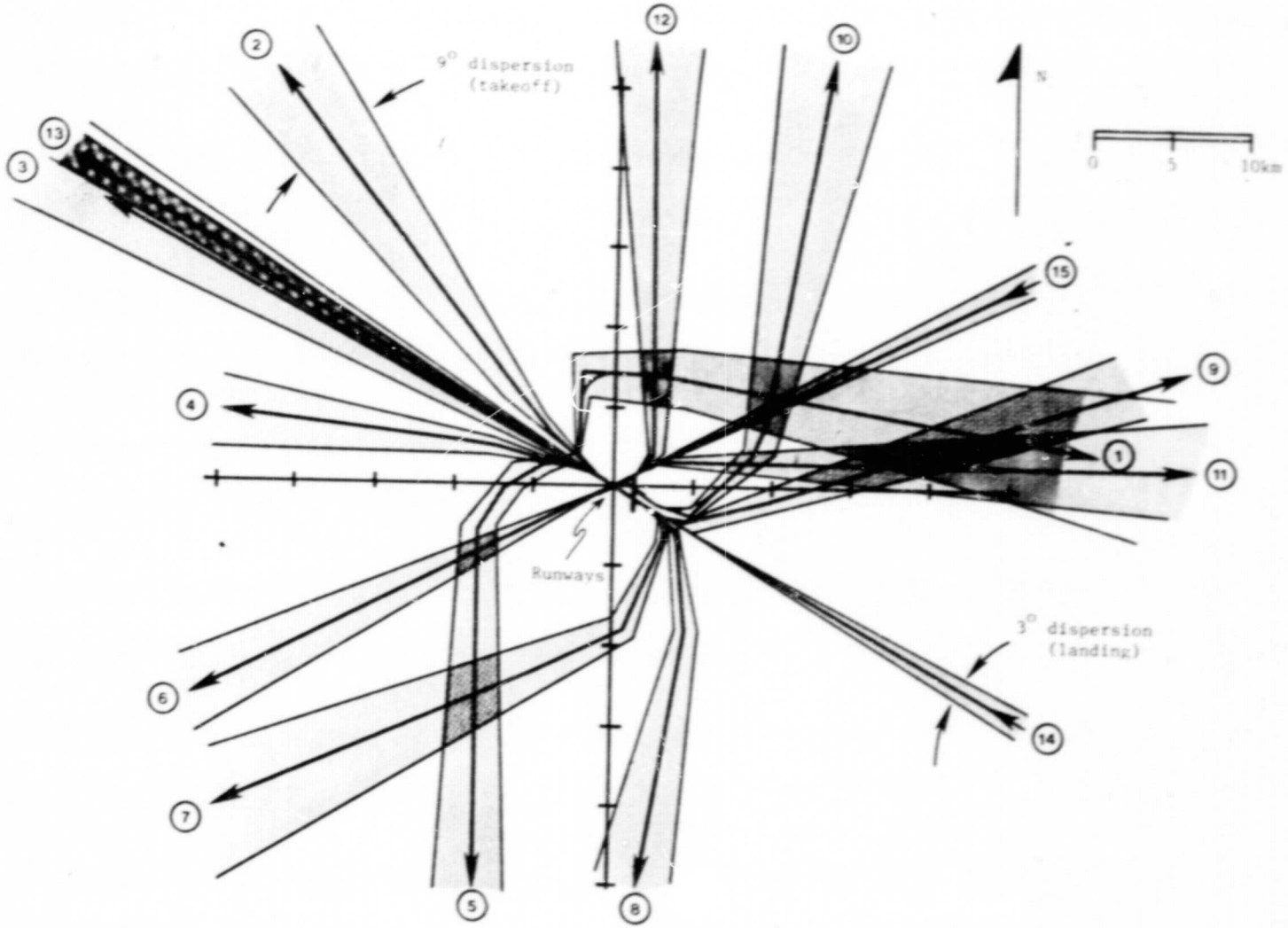


Figure 7.7a Lateral Dispersion in the Nominal Flight Paths at Lambert-St. Louis International Airport

starting points (representing significantly different trajectory sets) are used to perform the optimization a number of times. The lowest valued local minimum is then taken to be the global minimum.

Choices for the starting point are limited by two factors: the lateral dispersion on these paths and the constraints (particularly the aircraft dynamic and passenger comfort restrictions). For example as shown in Figure 7.7b, there are only five choices for track #7 starting points. Moving track #7 farther to the east would cause it to overlap track #8, in which case the two paths would be replaced by one. Consideration of cases farther to the north involves the same result with track #3. Thus, the presence of many trajectories limits the required amount of global searching, an important saving since the amount of computation needed for the optimization (local) increases with the number of flight paths. At smaller airports, where the number of trajectories is small, more starting points will be allowed, but the local optimization will require less computation.

In situations where the search has converged to a local minimum not significantly different from the starting point (but the existence of one is suspected), restarting the optimization after perturbing the corner points by some amount (~1,000 m.) may free the search from the region of the local minimum. Figure 7.8 shows an example of such a situation using a fictitious population distribution. Here, the short-range noise effects, as before, create a local minimum for the nominal landing path directly over the two lower population clusters. Repeated perturbation of the y-coordinates of the corner points eventually allows the optimization to find the global minimum. It is well outside of the

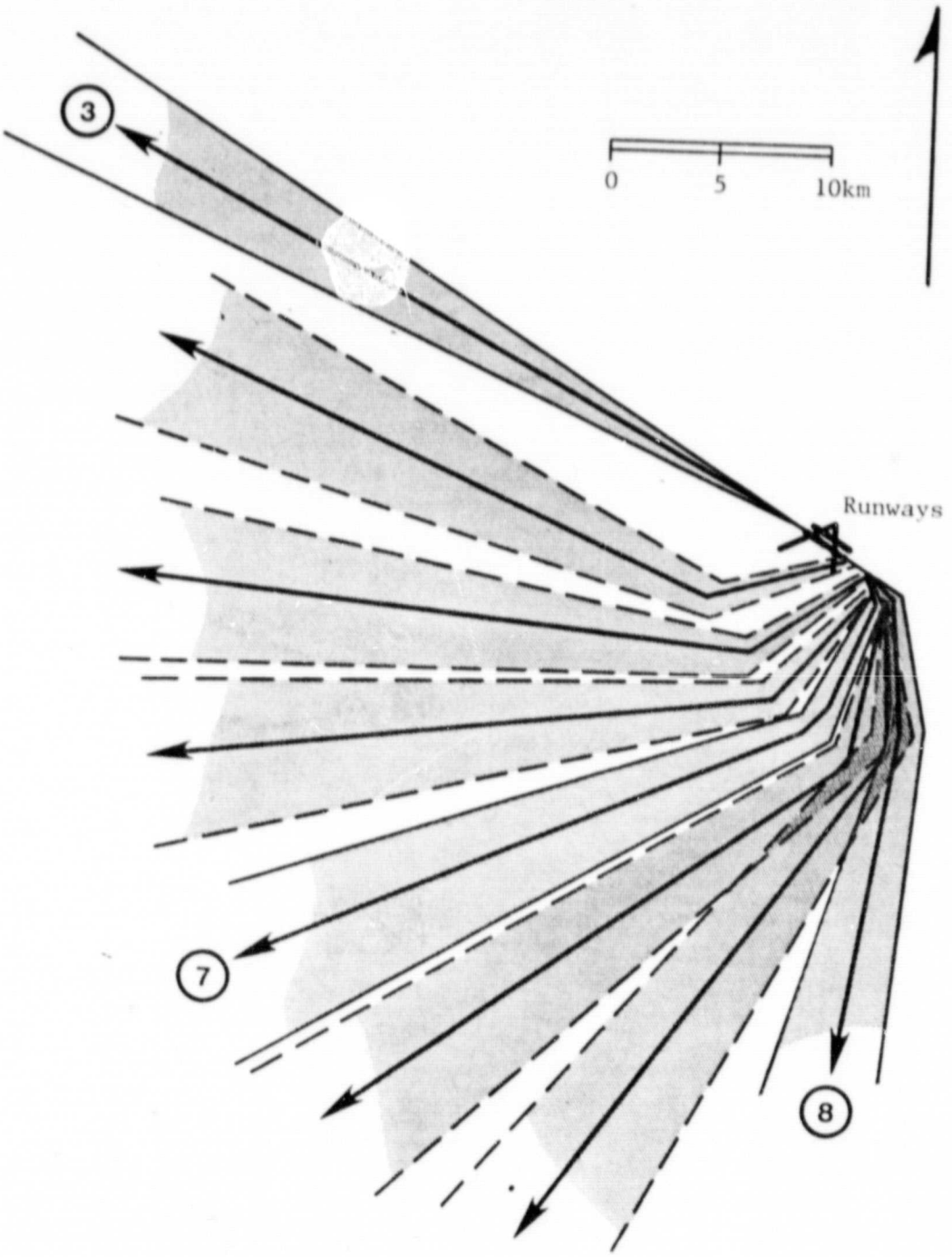


Figure 7.7b Starting Points for Track #7

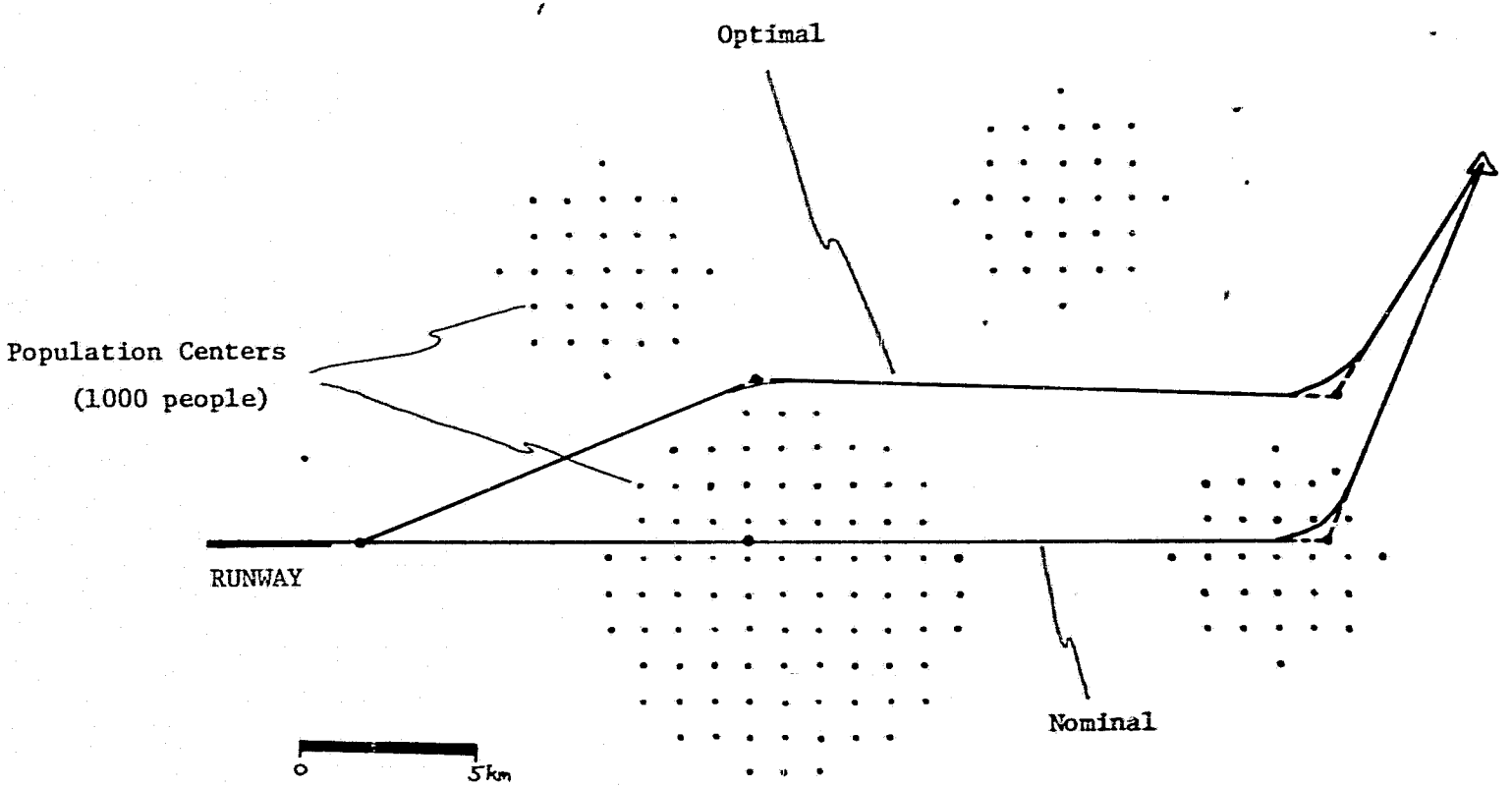


Figure 7.8 Local Minima in a Fictitious Population Distribution

dispersion limits of the nominal path and the reduction in NII' is 39%. The simplicity of the contrived problem provides for an easy decision about which way to perturb the corner points. In a real case, the direction may not be as obvious.

CHAPTER VIII

RESULTS

Two computer programs for implementing the system described in Chapter VII have been developed. The first, MANIP2, is a revision of MANIP (see Ref. [26]), with the following refinements: 1) optimization of a flight path's vertical profile is included, 2) takeoff and landing trajectories may be optimized, simultaneously, 3) a larger variety of commercial aircraft noise data is included, and 4) fuel and time of flight have been included in the cost function (as described in Chapter VI). The second program, CYCLIC, has the same features as MANIP2, except that 1) it employs the equivalent aircraft concept for computing noise levels, 2) the optimization scheme is modified to reduce storage requirements for large airports (i.e., many trajectories), and 3) fuel and time of flight are not included in the cost. A more complete description of CYCLIC is given in the Cyclic Optimization section of this chapter. Both programs were written in FORTRAN IV for use on CDC Cyber 170 series machines. Appendix D contains the source code listings.

The results of several investigations appear in the following sections. Two airports were used in these studies: Phoenix Sky Harbor International and Lambert-St. Louis International. Their respective population distributions are depicted in Figures 8.1a and 8.1b. In each grid section in Figure 8.1a (Phoenix) the number shown is the population count for that sector; smaller cell sizes in the St. Louis data necessitate (for legibility) the use of population density shading in Figure 8.1b. The body of results is divided as follows: 1) takeoff

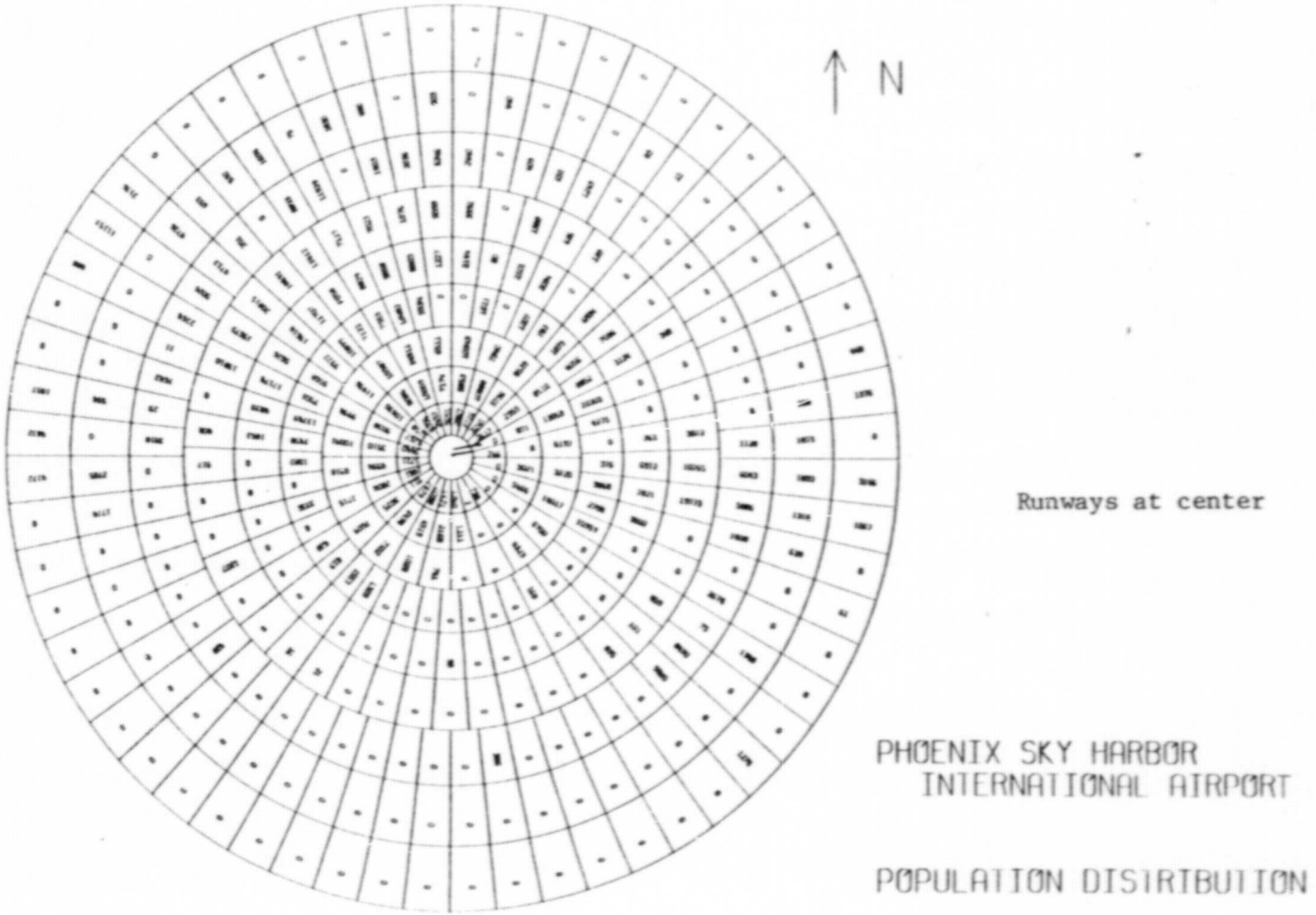


Figure 8.1a Phoenix (Sky Harbor International Airport) Population Distribution

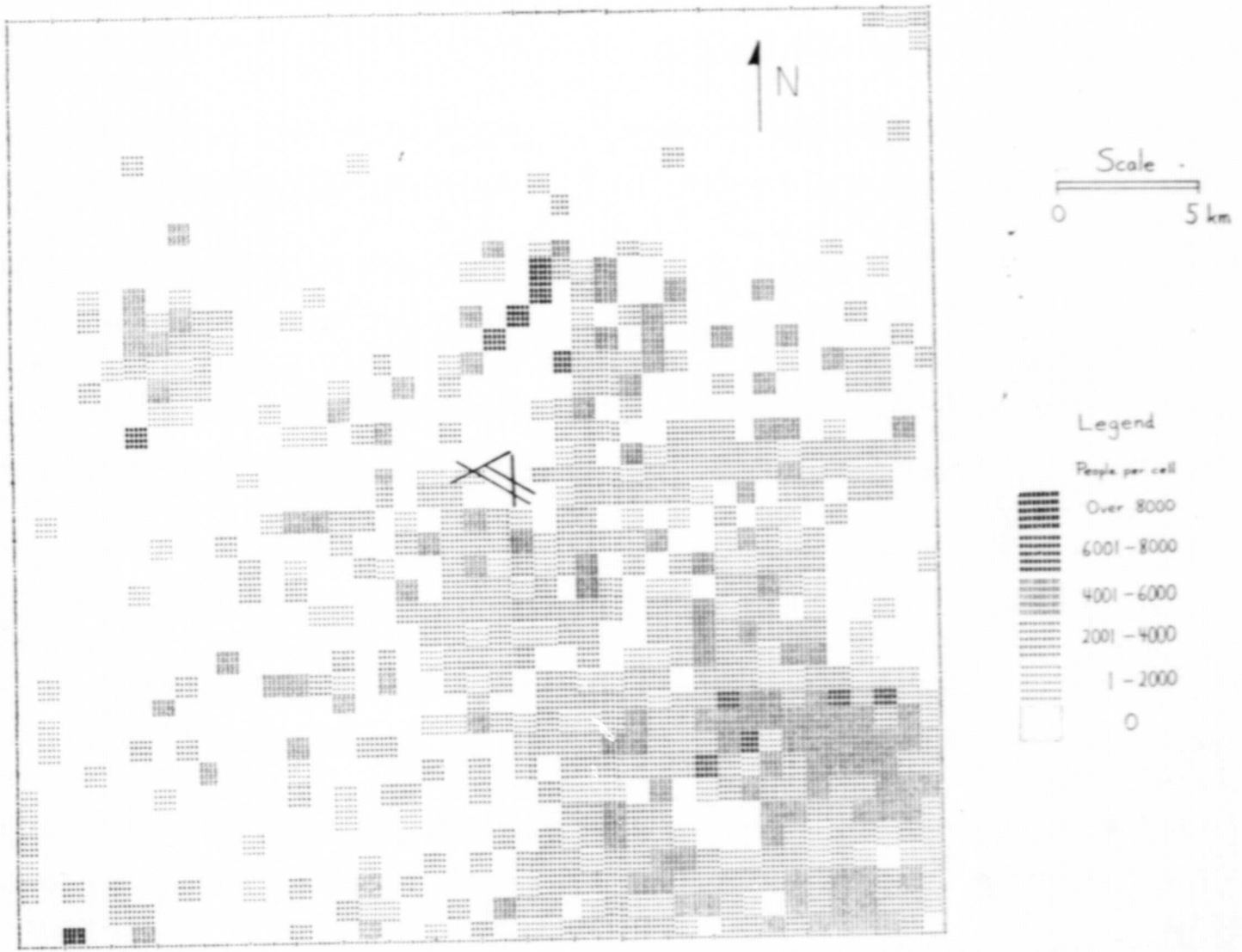


Figure 8.1b St. Louis (Lambert-St. Louis International Airport) Population Distribution

trajectories, 2) a comparison of optimum landing paths using NII, LWP, and HAPN as the annoyance criterion, 3) many takeoff and landing paths -- cyclic optimization, 4) a study of the effects of including the fuel and time of flight costs in the total cost function, and 5) the role of the population distribution in determining the optimum paths.

In all of the following sections, the flight paths are represented by their ground tracks, with the flight path angles, γ_c-climb, and γ_d-descent, indicated for each segment.

Takeoff Trajectory

Using the Lambert-St. Louis Airport community as a test case, a single trajectory was optimized[1], with LWP as the annoyance criterion. Both the nominal and optimum paths as shown in Figure 8.2. The nominal path was approximated by a five segment trajectory, with a 5° flight path angle (γ); the optimum path was restricted to three segments and $\gamma_{max} = 6°$. Nearly all of the variation between the two paths is in the ground tracks. This phenomenon, seen throughout the results that follow, occurs because 1) variations in the population density distribution occur only in the ground plane, and 2) the average distances from the flight paths to the population centers change more rapidly due to horizontal variations in the flight paths than due to vertical variations (owing to the small angles between the ground and the trajectories). The exception to this is, of course, when a segment

[1] Aircraft-trajectory assignment data for St. Louis were provided by the NASA-Langley Research Center [38].

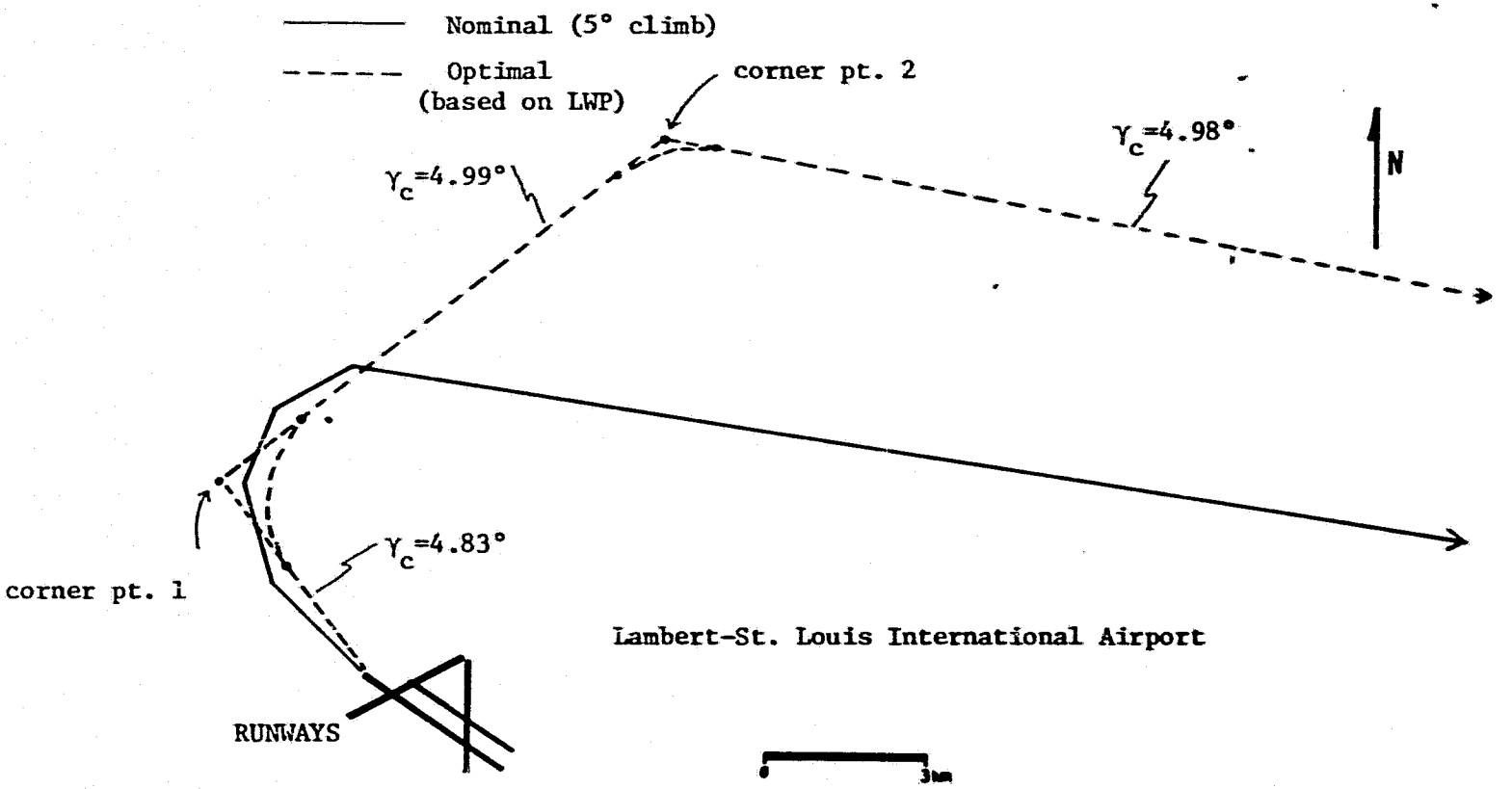


Figure 8.2 Single Takeoff Trajectory - Nominal and Optimum

near the runway (and closer to the ground) passes close to a population center. Then an increase in the flight path angle γ will produce a rapid decrease in the noise level at that center.

A point approximately 90 km. east of the airport was chosen as the outermost endpoint. This allowed the position of corner point #2 to vary without being restricted by the geometric constraints. The comparison of annoyance measures is as follows: nominal-LWP = 2571.1, NII = 0.161; optimum-LWP = 1267.9 (50.6% lower than nominal), NII = 0.228 (42% higher than nominal). This increase in NII, while LWP is decreasing, is another recurring phenomenon. It is caused by the change in total population used to normalize NII (i.e. people receiving $L_{dn} \geq 55dB$).

Comparison of Annoyance Models

In Chapter III, it was shown that for a simple population distribution (2 concentrated clusters of people), and a straight-line flight path, the measures NII and LWP indicated significantly different optimum positions for the path. Here, NII, LWP, and HAPN will be compared for general trajectories and a real population distribution (Phoenix Sky Harbor Airport community). This section also serves to illustrate the calculation of optimum landing paths.

For landings on the runway indicated in Figure 8.3a there are two entry points (initial approach fixes). This figure shows the nominal ground tracks; the glide slopes are 3°. An estimate of the mixture of aircraft types operating at this airport was obtained from data in the Official Airline Guide [39]. Because the nominal paths are joined

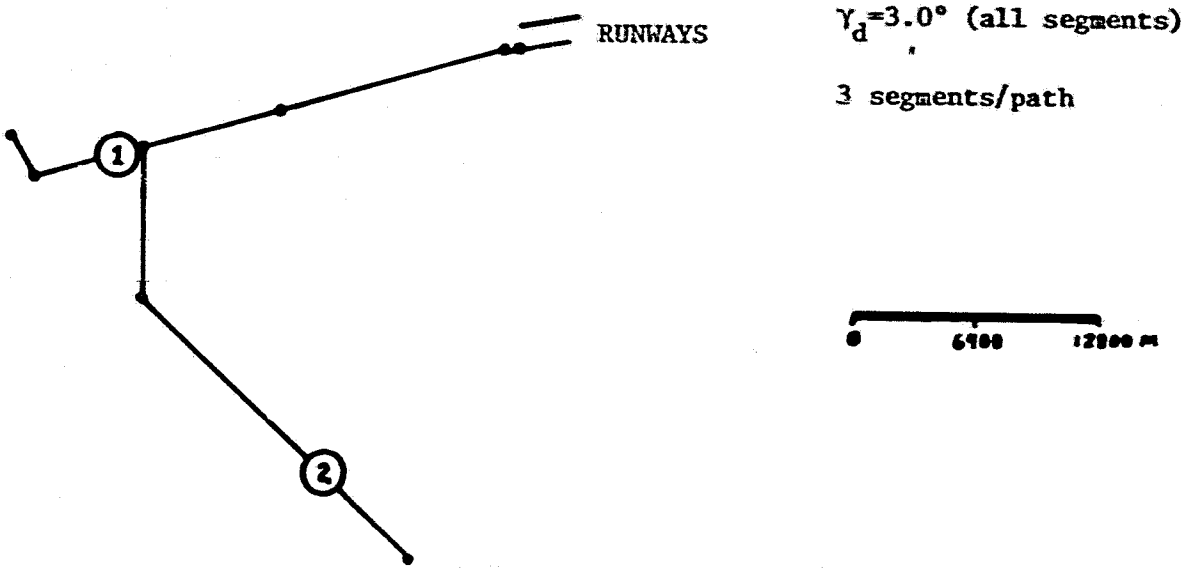


Figure 8.3a Nominal Trajectories at Phoenix Sky Harbor Airport

together over a large portion of their lengths, it is assumed that separation for safety purposes is not a requirement for traffic control here; therefore the separation constraint is bypassed. Note that there are three segments on each path, with some overlapping.

With NII as the annoyance measure, the optimum pair of trajectories obtained is that shown in Figure 8.3b. Values of NII, LWP and HAPN for this pair are given in Table 8.1, along with comparisons with the corresponding nominal values.

With either LWP or HAPN as the annoyance measure, the optimum pair is radically different, as seen in Figure 8.3c. The annoyance measures associated with these flight paths are given in Table 8.1. It is seen that the improvement in the annoyance based upon LWP or HAPN is greater than that of NII. Further, NII increases when LWP and HAPN decrease, and vice versa, another indication of the normalization problem in NII.

Fuel and Time of Flight Study

As an example of how the system may be used to study associated problem, the cost of fuel and time of flight ($c_{F,TOF}$) has been incorporated in the total cost function. In this investigation, the relation between $c_{F,TOF}$ and NII is examined.

Using the modified cost function, Eqn. (6-13), the optimization was repeated for increasing values of $w_{F,TOF}$ (the fuel and time of flight weighting factor); the airport used was Phoenix Sky Harbor. Figure 8.4a summarizes the results, showing the relation between the NII, and the fuel and time of flight cost ($c_{F,TOF}$). Accompanying it are Figures

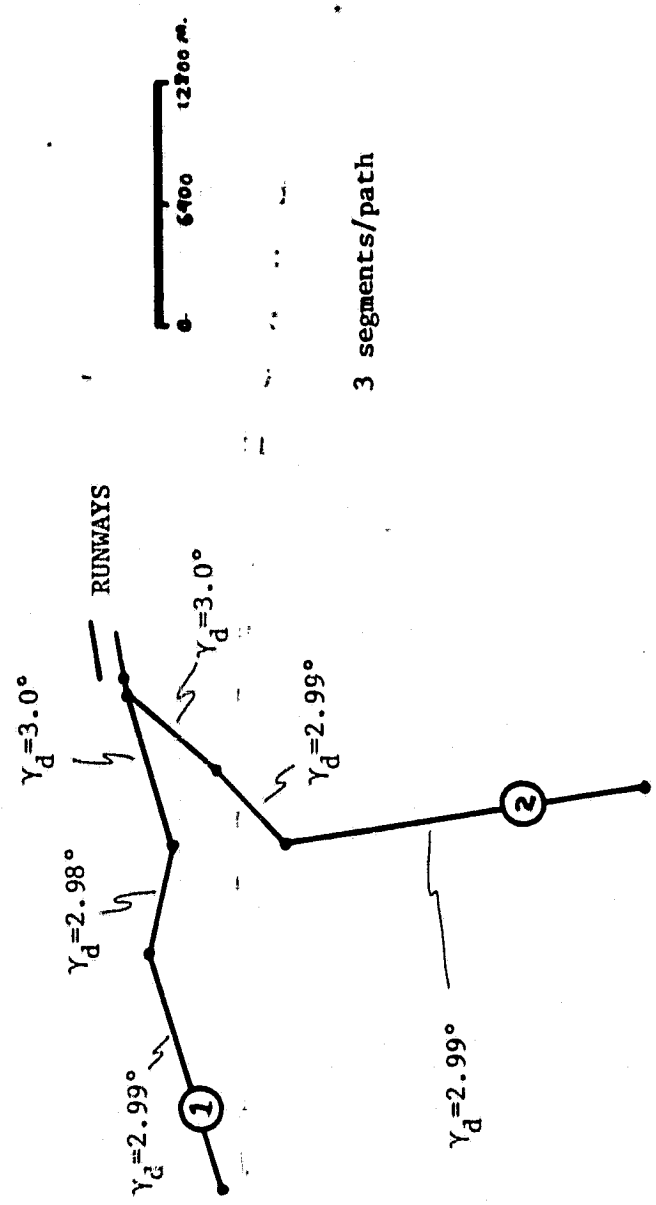


Figure 8.3b Optimum (NII) Trajectories at Phoenix Sky Harbor Airport

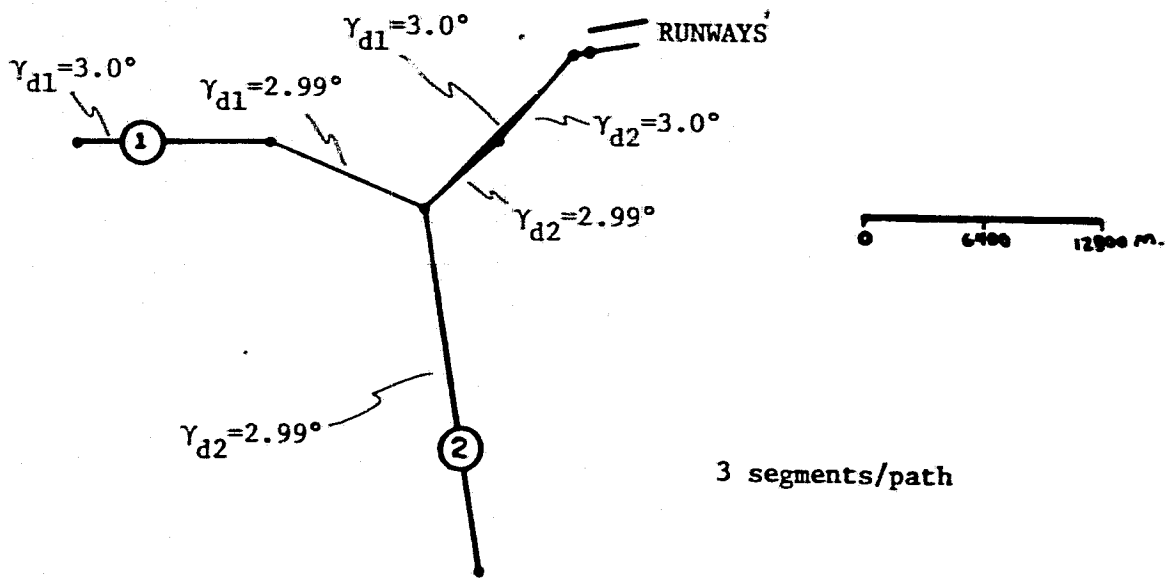


Figure 8.3c Optimum (LWP or HAPN) Trajectories at Phoenix Sky Harbor Airport

Table 8.1

Comparison of Annoyance Measures
(For Phoenix Sky Harbor)

Trajectory Set	NII	†Δ%	LWP	†Δ%	HAPN	†Δ%	Figure Showing Optimum
Nominal	0.281		30566		0.491		8-3a
NII optimum	0.264	-6	31993	4.7	0.524	6.7	8-3b
LWP optimum	0.290	3	25192	-17	0.445	-9.3	8-3c
HAPN optimum	0.290	3	25192	-17	0.445	-9.3	8-3c

†Percentage difference from nominal value.

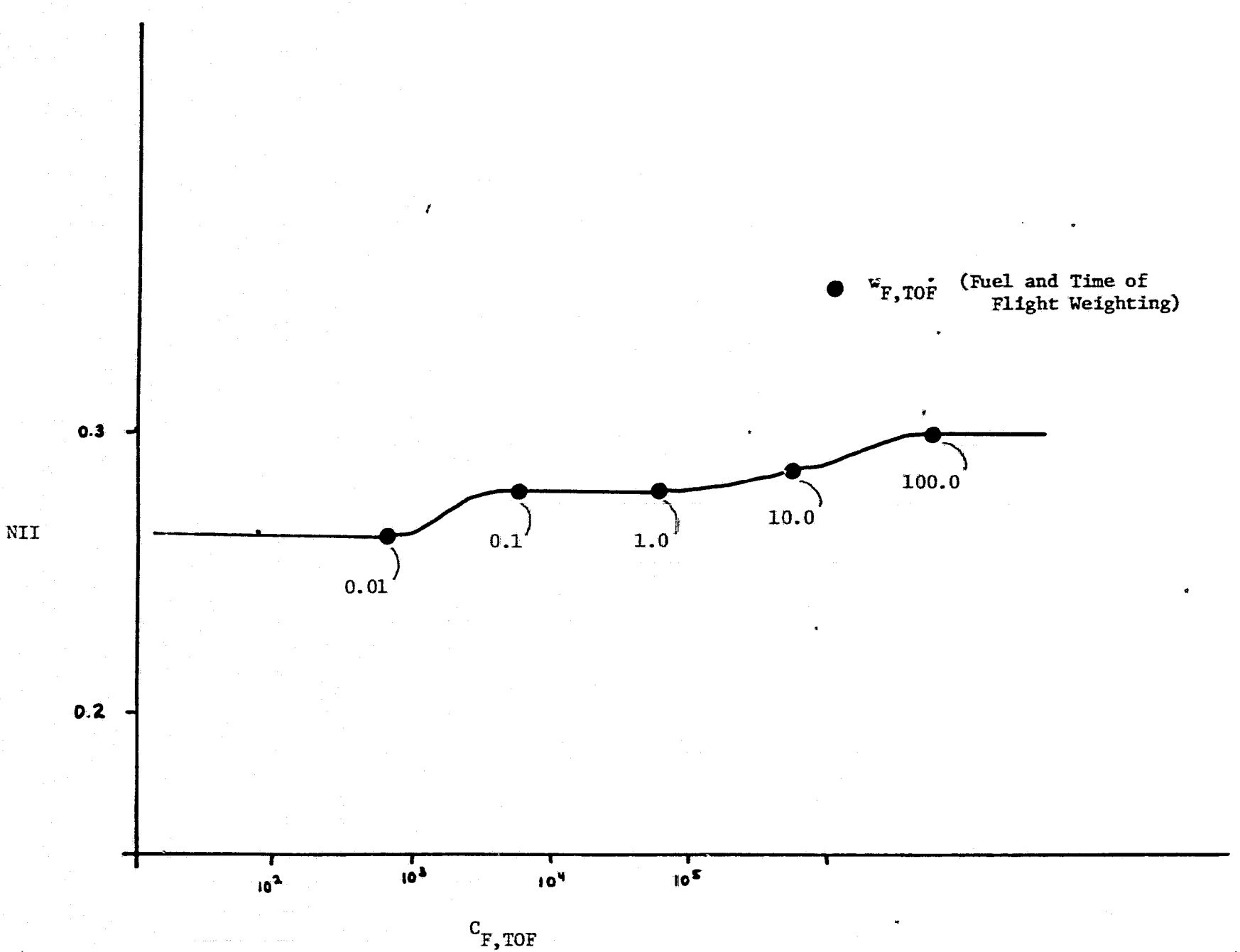


Figure 8.4a NII vs. Fuel and Time of Flight Cost ($c_{F,TOF}$)

8.4b through 8.4f, which show the optimum landing trajectories for each value of $w_{F,TOF}$ used.

As expected, the flight paths become shorter (straighter) as $c_{F,TOF}$ increases. Trajectory #1 shortens rapidly ($w_{F,TOF} = 0.1$) to a straight path, but trajectory #2 is hindered by two factors: a large cluster of people southeast of the runways (see Figure 8.1a), and the final angle of approach restriction. For $w_{F,TOF} \leq 3$ shortening this path will result in NII increases that are greater than any decrease in the fuel and time of flight cost; hence, there is a flat region in Figure 8.4a for $0.1 \leq w_{F,TOF} \leq 3$. Larger values of $w_{F,TOF}$ increase the relative importance of $c_{F,TOF}$, and the total cost is reduced at the expense of NII. It is impossible, though, for trajectory #2 to become completely linear because of the final angle constraint. The effect of $c_{F,TOF}$ is then saturated at $w_{F,TOF} = 100$. The extrapolation for very small $w_{F,TOF}$ is based upon the value obtained for NII in the Comparison of Annoyance Measures section (where $w_{F,TOF} = 0$).

Cyclic Optimization

At large airports such as Lambert-St. Louis International, the large variety of aircraft that must be considered necessitates the use of the equivalent aircraft concept in order to keep the computation time manageable. Further, the number of trajectories being optimized (15 at St. Louis) will require a considerable increase in storage requirements, compared with those needed for Phoenix. If the storage required is in excess of the allocated resources, the dilemma may be remedied by optimizing the flight paths one at a time, cyclically. The power density

Figure 8.4b Optimum Trajectories, $w_{F,TOF} = 0.01$

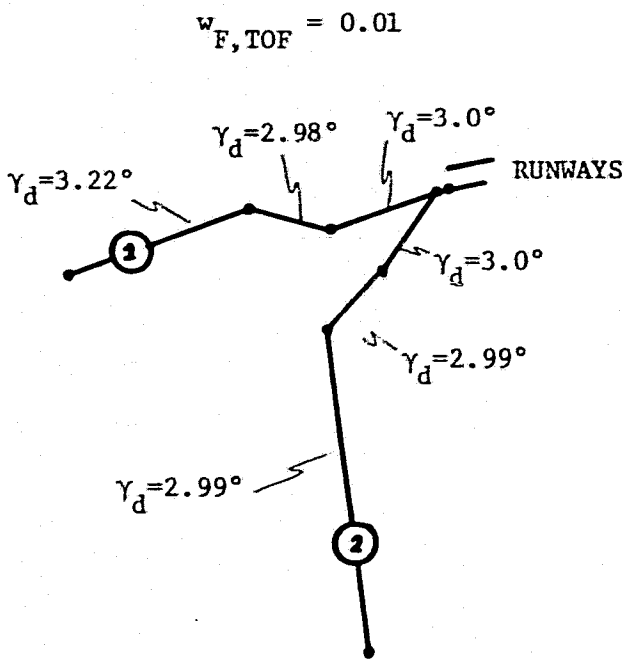


Figure 8.4c Optimum Trajectories, $w_{F,TOF} = 0.1$

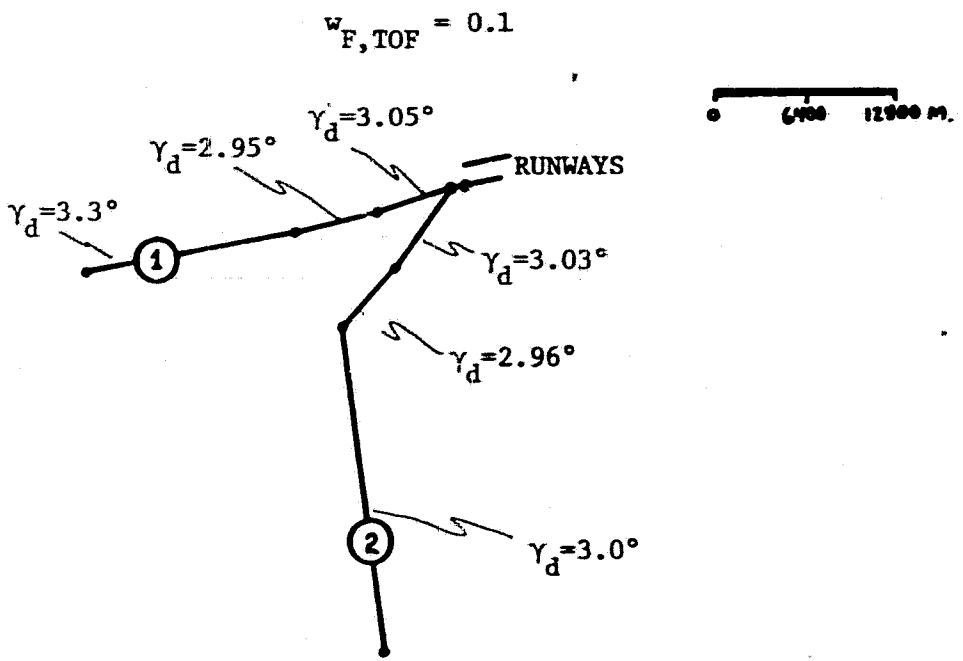


Figure 8.4d Optimum Trajectories, $w_{F,TOF} = 1.0$

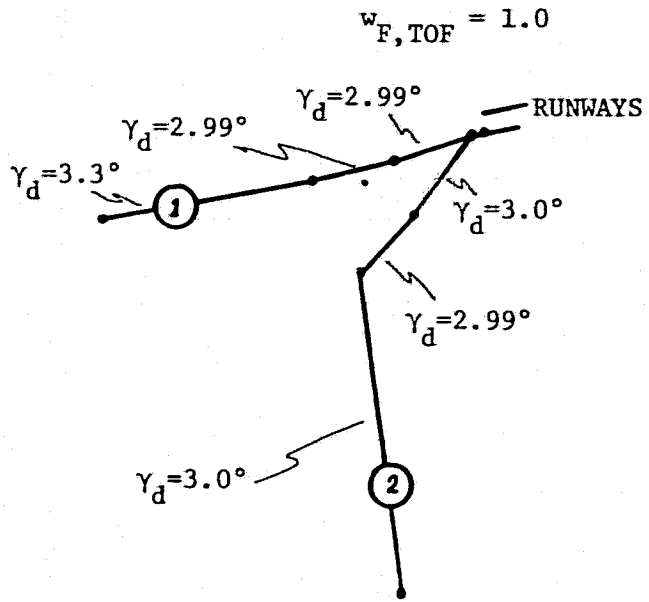
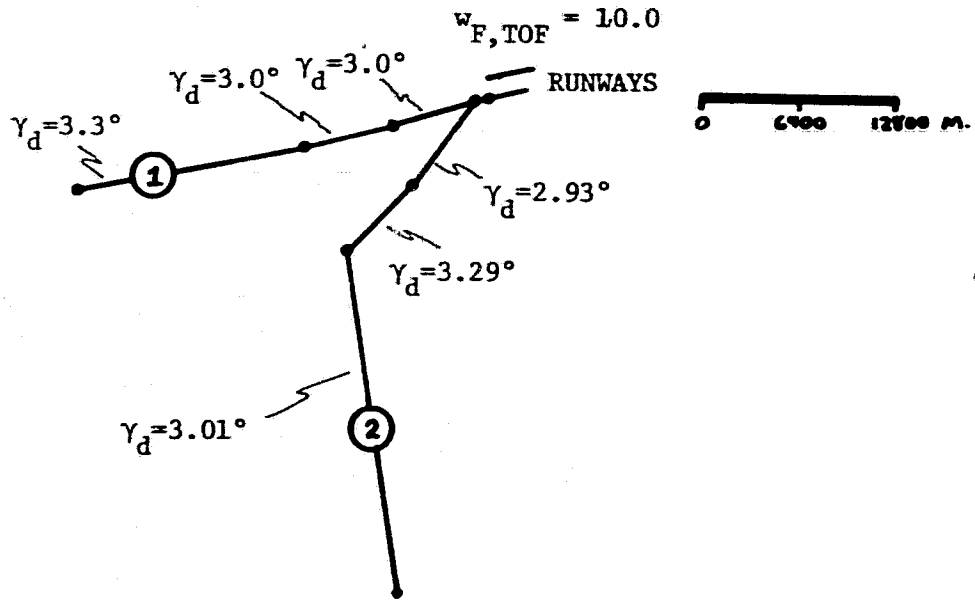


Figure 8.4e Optimum Trajectories, $w_{F,TOF} = 10.0$



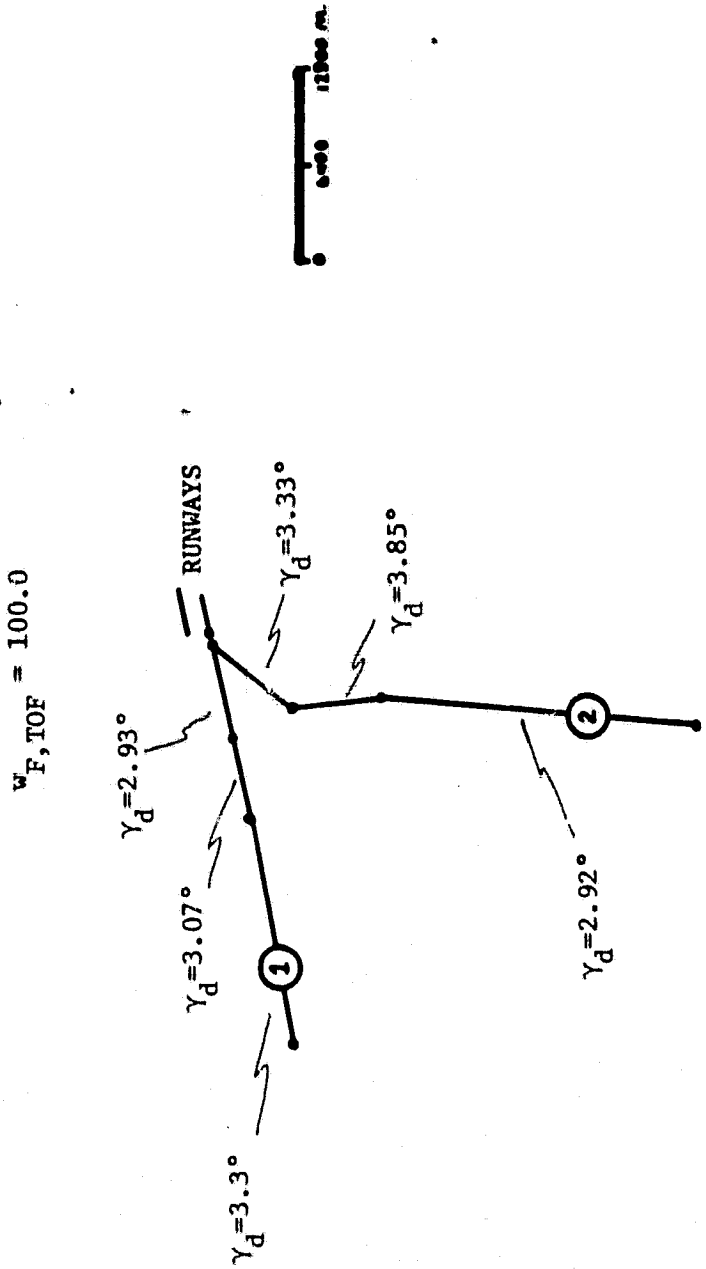
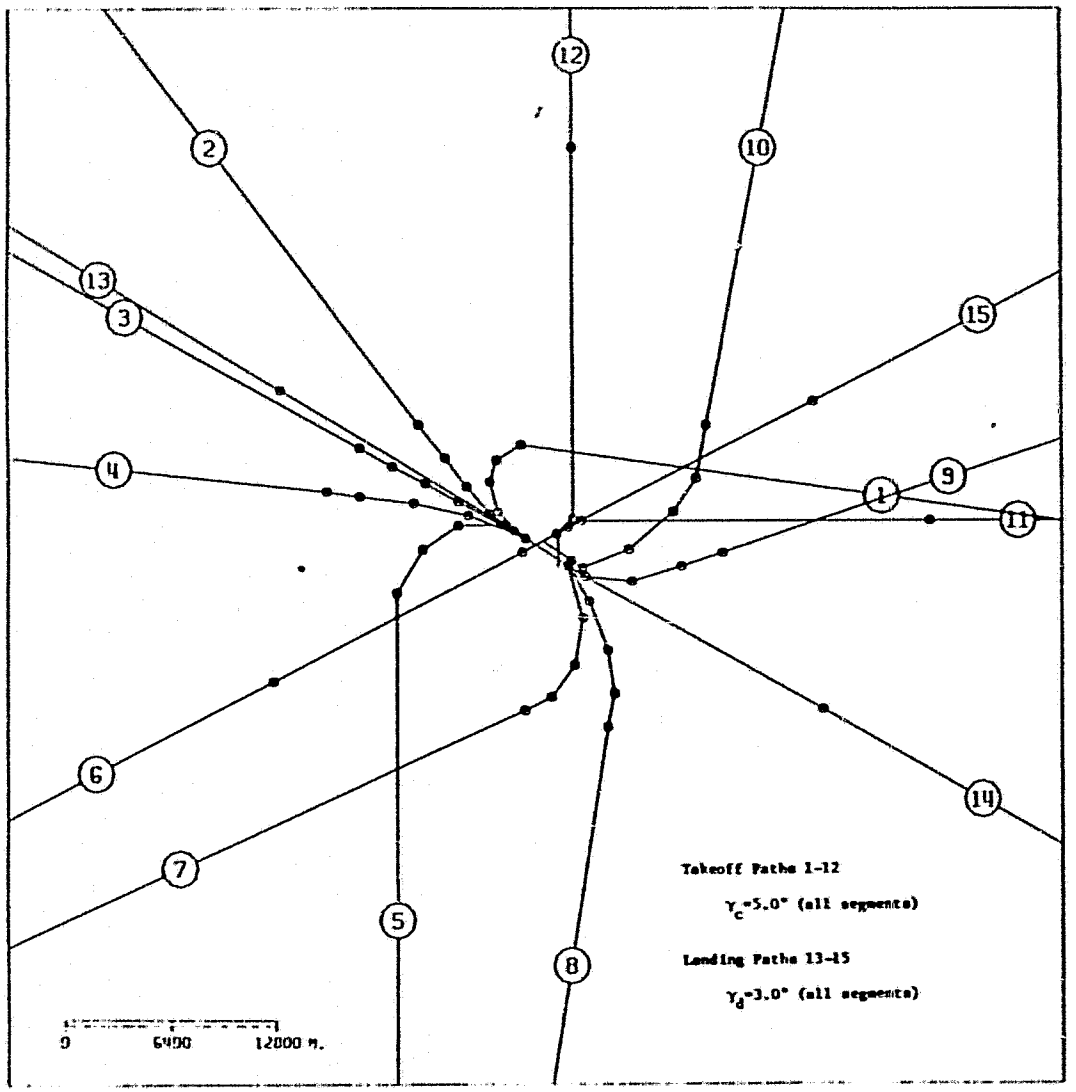


Figure 8.4f Optimum Trajectories, $W_{F,TOF}$ = 100.0



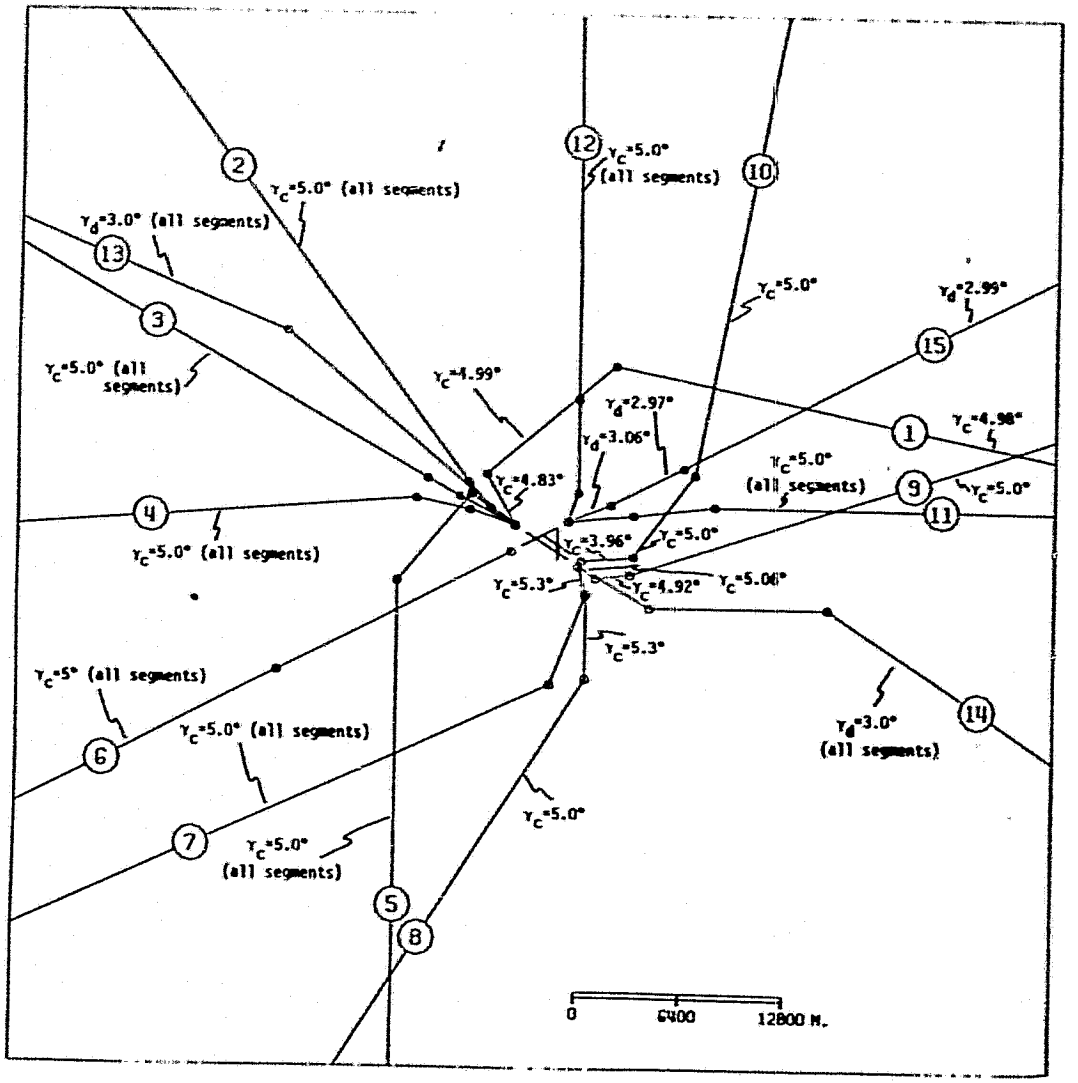
St. Louis Nominal Trajectory Set

5 segments/trajectory

NII = 0.2076

LWP = 105230

Figure 8.5a St. Louis Nominal Trajectory Set



St. Louis Optimum Trajectory Set

3 segments/trajectory

NII = 0.2108

LWP = 77849

Figure 8.5b St. Louis Optimum Trajectory Set

contributions from the (temporarily) constant trajectories are stored and then summed with those of the path being optimized in order to determine the L_{dn} value in each grid cell. As a result, the optimization storage requirements are for only one trajectory, but the annoyance measure is for the entire set of paths.

It should be noted that this approach does not guarantee convergence to a minimum; however, the results for the St. Louis case are excellent. Figure 8.5a shows the nominal paths at Lambert-St. Louis International (tracks 1-12: takeoff, 5°; track 13-15: landing, 3°). Each trajectory is approximated with 5 segments. Judging by the population distribution (Figure 8.1b), it was decided that paths 1,2,7, 8,9,13,14, and 15 could give the most improvement in the annoyance (LWP). Using the program CYCLIC to cyclically optimize these paths (see Appendix D), and allowing a maximum of three segments per trajectory, the results obtained were: LWP decreased 26%, NII increased 1.5%, from the nominal vlaues. Figure 8.5b shows the improved trajectories.

Population Distribution and Aircraft Mix

In order to investigate the relative importance of the population distribution vs. the aircraft mix (i.e., mixture of different aircraft types) on the optimum flight paths, a study was conducted using two different population distributions (one fictitious, one real), and two different aircraft mixes.

Using the fictitious population distribution shown in Figure 8.6, a single optimum landing trajectory (3°) was computed, using the CYCLIC

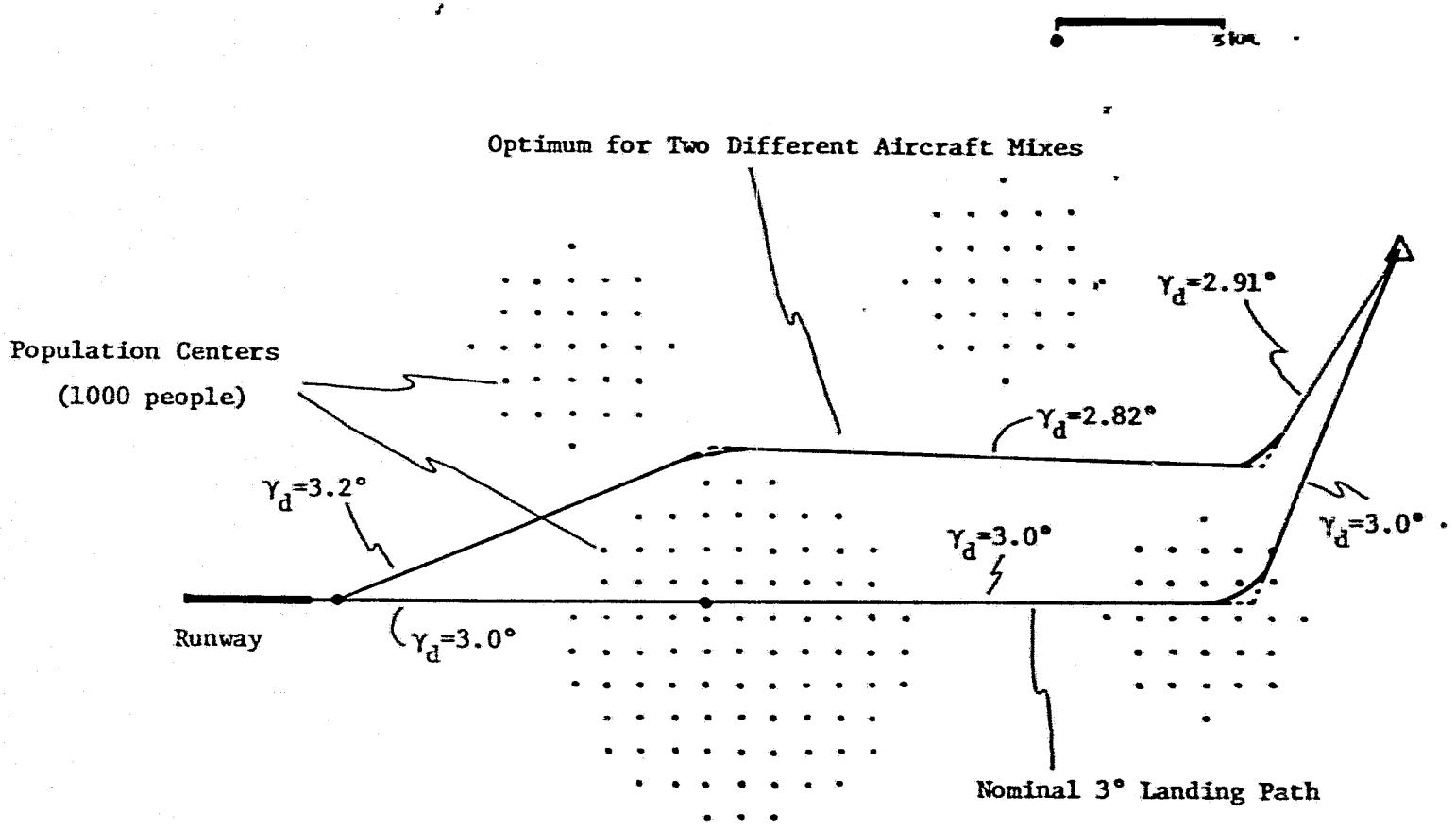


Figure 8.6 Fictitious Population Distribution - The Effect of Different Aircraft Mixes

program,[2] Two different mixtures of aircraft types were investigated; the degree of difference is evinced by the difference in their equivalent aircraft noise coefficients:

Mix #1: $c_{1,eq} = 170.70$ $c_{2,eq} = 24.23$

Mix #2: $c_{1,eq} = 166.00$ $c_{2,eq} = 28.60$.

The optimum paths for the two mixes were nearly identical; no two corresponding corner points differ by more than 36 m.

With a real population distribution, though, a strong dependence upon the aircraft mix is evident. For the Phoenix airport, two mixes were examined:

Mix #1: $c_{1,eq} = 169.47$ $c_{2,eq} = 24.37$

Mix #2: $c_{1,eq} = 166.00$ $c_{2,eq} = 28.60$,

Mix #1 corresponds to the true mix at Phoenix, which was used in the Comparison of Annoyance Models section and for which the optimum results appear in Figure 8.3b (NII) and 8.3c (LWP). Mix #2 is all DC-10 aircraft, with the equivalent amount of passenger seating of Mix #1. In the case of Mix #2, the optimum paths are significantly different, with either LWP or NII as the annoyance criterion. Figure 8.7a and 8.7b depict the results.

The consequence of this example is that the relative importance of the population distribution vs. the aircraft mix in determining the optimum trajectories cannot be stated in general. For a simple distribution of people, corridors for optimum trajectories that are

---

[2] Although only one trajectory, with three segments, was being optimized, CYCLIC was used so that the equivalent aircraft concept could be employed.

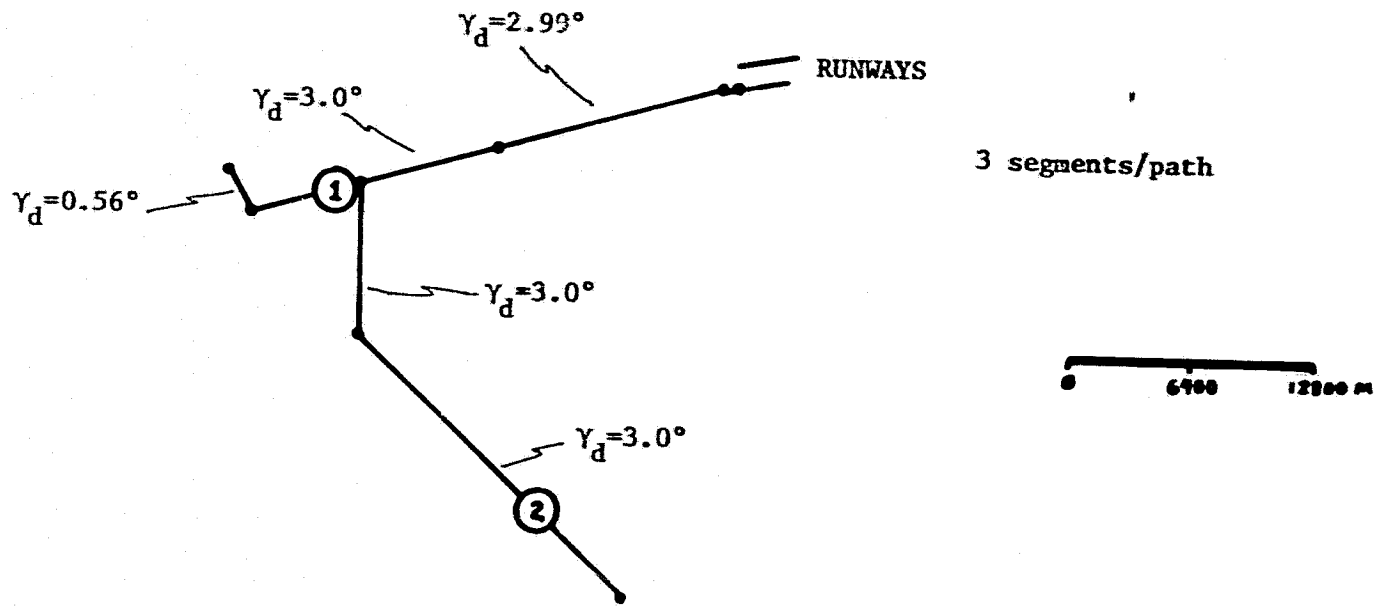


Figure 8.7a Optimum (NII) Landing Trajectories at Phoenix Sky Harbor Airport (DC-10 mix)

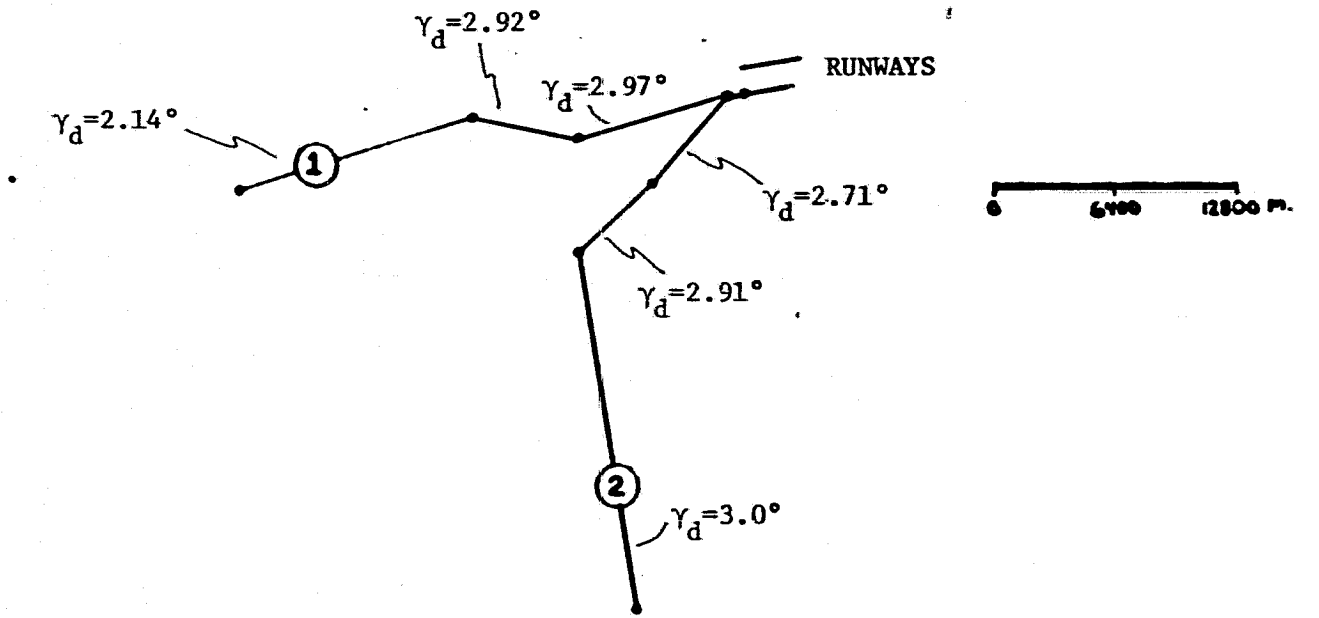


Figure 8.7b Optimum (LWP) Landing Trajectories at Phoenix Sky Harbor Airport (DC-10 mix)

independent of aircraft mix may exist (such appears to be case in the fictitious distribution in this example). A real population distribution, though, is usually more dispersed, which results in more people (on the average) being closer to the flight paths. Because of the nonlinearities in the noise and annoyance models, changes in the aircraft mix result in a change in the annoyance of unpredictable magnitude. Therefore, new optimum flight paths must be computed when the mix changes.[3]

[3] For a particular situation, however, experience may indicate that small changes in the mix do not necessitate recomputation of the trajectories.

CHAPTER IX

CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER STUDIES

This study has developed a method to assess the ability to reduce, through operational controls, the annoyance caused by aircraft noise in a community. Test cases, using a computer implementation of the method and actual airport situations, have shown the feasibility of using trajectory modifications in the solution to the problem of aircraft noise. The amount of reduction that can be gained depends upon the airport, aircraft mix, constraints, and model of annoyance being used.

A comparison study of the annoyance measures NII, LWP, and HAPN has shown that, in addition to resulting in different percentage improvements, the optimum trajectories that result from using each of these measures can be significantly different. Because of its mathematical definition, NII has been shown to be an inferior indicator of the amount of change in annoyance for a given community.

The method has a modular structure which allows for straightforward modification of any of its components, e.g., the flight path model, the aircraft noise model, the optimization algorithm, etc. In addition, the method may be expanded to include related problems in the optimization. A particular example of this was demonstrated by including the "costs" of fuel and time of flight in the cost function. The modified system was then used to examine the influence of these factors (along with the annoyance) on the resulting optimum flight paths.

Of particular importance is the finding that the population distribution in a given community does not necessarily determine a unique set of optimum trajectories; the mixture of aircraft strongly influences the solution. This is unfortunate from a computational aspect, since the mixtures on the flight paths generally will vary over time, requiring that new optimum sets be calculated.

Two concepts that may have applications elsewhere were developed in this work. First, the complexity of computing the flight paths was reduced by assuming a parametric solution and explicitly including restrictions based upon analyses of the linear dynamic equations of motion of the aircraft. This technique may be useful in situations where the computing resources available prohibit the customary variational approach to flight path optimization. Second, the equivalent aircraft concept not only reduces the time required for calculation of noise levels due to multiple sources, but also provides a means for predicting noise levels as a result of changes in the number of aircraft on any trajectory.

The difficulty associated with searching over many local minima was overcome by making use of the statistical lateral dispersions in the trajectories. These dispersions limit the size of the alternative regions (in the horizontal plane) in which to search for significantly different optimum paths.

Finally, this approach to the problem of aircraft noise has shown that annoyance can be reduced significantly via operational control. The method provides a tool for those who must decide what action to take in remedying the problem. It can also be used to optimally site an airport in a community to minimize future noise problems.

Recommendations

It is felt that further work should employ a population distribution model that does not artifically creat optimum pathways. An analytic representation of the distribution (e.g. spline functions, or low order polynomials) would likely aid in the operation of the method, without necessarily increasing computational requirements.

Even with such a refinement, the difficulty of local minima in the cost function, caused by real, macroscopic variations in the population density will persist. Use of the statistical lateral dispersion in flight paths has been instrumental in restricting the global search and should be automated.

The assumption of constant runway designations, i.e., takeoff or landing, which was made in this study, is not generally true. Future work should include the possibility of having to compute an alternate set of flight paths, perhaps because of a change in the wind direction during the day. The cumulative power density distributon from the previous optimum configuration would have to be retained and combined with that of the new set in order to include the total effect over the period of time in question. It is envisioned that air traffic controllers could then use this method to compute alternate optimum flight paths in real time.

Finally, community land use planning could employ this system by restricting the maximum noise levels in proposed sensitive areas (schools, hospitals, etc.). The cost function would be adjoined by penalties for violations of these restrictions. Industrial areas where high noise levels from aircraft would have little effect, would be

weighted less heavily in the annoyance calculation. As the final step, the coordinates of the airport centroid would be included as unknown parameters to be determined (with some restrictions) in the optimization procedure. The result would be not only the optimum set of trajectories, but also an optimally positioned airport.

APPENDIX A

SPECIAL CONSIDERATIONS FOR THE NOISE LEVEL CALCULATIONS

Energy Addition

Sound intensity (power/area) is usually expressed in the form

$$L = 10 \cdot \log_{10} \frac{I}{I_0} \tag{A-1}$$

where I is the intensity being characterized and I_0 is a reference intensity (usually taken as the lowest audible intensity at 1,000 Hz, approximately 10^{-16} watt/cm^2). The units are called decibels (dB) and provide a convenient scale for the extremely wide range of intensities that may occur.

For a given volume of space (in which the intensity of sound is constant), the acoustic energy received is proportional to the intensity and the time of exposure. More precisely,

$$E = aI\Delta t \tag{A-2}$$

where Δt is an increment of time during which I does not change appreciably, E is the increment of energy received, and a is a constant that depends upon the volume of space being considered. The sound level may then be expressed as

$$L = 10 \cdot \log_{10} \frac{E}{E_0} \tag{A-3}$$

and

$$E = E_0 \cdot 10^{L/10} \tag{A-4}$$

where E_0 is the energy received from the reference sound.

This provides a powerful tool for "adding" sound from different sources. Suppose that n different sources produce sound with levels L_i at a particular location. The total acoustic energy per unit volume of space is

$$E_{TOT} = \sum_{i=1}^{n} E_i = E_0 \sum_{i=1}^{n} 10^{L_i/10} \qquad \text{(A-5)}$$

Eqn. (A-3) is used to calculate the intensity level in dB:

$$L_{TOT} = 10 \cdot \log_{10} \sum_{i=1}^{n} 10^{L_i/10} \qquad \text{(A-6)}$$

In Chapter III, the L_{dn} measure uses the energy addition method to obtain a weighted average of noise levels occuring over a period of time. The use of energy addition is often referred to as adding on an "energy basis," although it is actually power densities that are being added.

Conversion of EPNL to L_A

The measure Effective Perceived Noise Level (EPNL) is based upon Perceived Noise Level (PNL) in the following way (see Ref. [5], pp. 287-300.):

$$\text{EPNL} = \text{IPNL} - 12 + (\text{onset corrections}) + (\text{impulse corrections}) \qquad \text{(A-7)}$$

where IPNL is the Integrated Perceived Noise Level

$$\text{IPNL} = 10\ 1\ \log_{10} \sum_{i=1}^{16} 10^{\text{PNL}_i/10} . \qquad \text{(A-8)}$$

The values of PNL are taken every 0.5 sec. for 8 sec., hence the upper limit of 16 in the summation.

Neglecting the onset and impulse corrections (they are not significant for aircraft noise) gives

$$\text{EPNL} \simeq 10 \cdot \log_{10} \left(\sum_{i=1}^{16} 10^{\text{PNL}_i/10} \right) - 12 \qquad \text{(A-9)}$$

$$\simeq 10 \cdot \log_{10} \left(\frac{\sum_{i=1}^{16} 10^{PNL_i/10}}{16} \right) \qquad \text{(A-10)}$$

The last expression is just the average PNL (on an evergy basis). So for a constant PNL,

$$EPNL \simeq PNL \qquad \text{(A-11)}$$

Finally, for nonimpulsive sound

$$PNL \simeq L_A + 13 \text{ dB}, \qquad \text{(A-12)}$$

giving

$$L_A \simeq EPNL - 13 \text{ dB}. \qquad \text{(A-13)}$$

APPENDIX B

THE LINE SEARCH

In Chapter V, several of the optimization methods require a one-dimensional, or line, search in each iteration. The task is to find the value of α that minimizes the cost function f, i.e.

$$\underset{\alpha}{\text{minimize}}\ f(\underline{x}_k + \alpha\hat{\underline{d}}_k)$$

where $\underline{x}_k$ is the current position (at the k-th iteration) in E^n and $\hat{\underline{d}}_k$ is a unit vector that defines the search direction, relative to $\underline{x}_k$.

A successful method for finding α, which has been employed in this work, is that of assuming a polynomial form in α for $f(\underline{x}_k + \alpha\hat{\underline{d}}_k)$, the exact minimum of which is determined by the calculus of one variable. The form used here is cubic, i.e.,

$$f(\underline{x}_k + \alpha\hat{\underline{d}}_k) = k_0 + k_1\alpha + k_2\alpha^2 + k_3\alpha^3, \tag{B-1}$$

for which the minimum occurs at α^*, such that

$$\left.\frac{\partial f}{\partial \alpha}\right|_{\alpha=\alpha^*} = k_{\pm} + 2k_2\alpha^* + 3k_3\alpha^{*2} = 0\ , \tag{B-2}$$

and

$$\left.\frac{\partial^2 f}{\partial \alpha^2}\right|_{\alpha=\alpha^*} = 2k_2 + 6k_3\alpha^* > 0\ . \tag{B-3}$$

The last equation eliminates one of the two roots of Eqn. (B-2), so that a unique minimum is given once the k_i are specified. This is accomplished by calculating f and $\frac{\partial f}{\partial \alpha}$ at two values of α. Denoting $f(\alpha_0)$, $f(\alpha_1)$, $\frac{\partial f}{\partial \alpha}(\alpha_0)$, and $\frac{\partial f}{\partial \alpha}(\alpha_1)$ by f_0, f_1, f_0', and f_1', respectively,

the minimum, α_2, of the cubic function is given by the following:[1]

$$u_1 = f_0' + f_1' - 3 \left[\frac{f_0 - f_1}{\alpha_0 - \alpha_1} \right] \tag{B-4}$$

$$u_2 = [u_1^2 - f_0' f_1']^{\frac{1}{2}} \tag{B-5}$$

$$\alpha_2 = \alpha_1 - (\alpha_1 - \alpha_0) \left[\frac{f_1' + u_2 - u_1}{f_1' - f_0' + 2u_2} \right] . \tag{B-6}$$

Calculation of the directional derivative $\frac{\partial f}{\partial \alpha}$ is normally performed by projecting the gradient $\underline{g}(\underline{x}_k) = \nabla^T f(\underline{x})$ onto the direction $\underline{d}_k$, i.e.

$$\frac{\partial f}{\partial \alpha} = \nabla^T f(\underline{x}) \cdot \hat{\underline{d}}_k , \tag{B-7}$$

where (.) denotes the inner product in E^n. Since derivatives of f must be approximated numerically in this work, though, it is simpler to use the definition of derivative to obtain

$$\frac{\partial f}{\partial \alpha} \simeq \frac{f(\underline{x}_k + \Delta\alpha \hat{\underline{d}}_k) - f(\underline{x}_k)}{\Delta\alpha} \tag{B-8}$$

where $\hat{\underline{d}}_k$ is a unit vector along $\underline{d}_k$ and $\Delta\alpha$ is a small increment.

Choosing values of α_0 and α_1 for use in the cubic fit presents no problems if f is cubic: the fit will be exact regardless of what values of α are used. With a non-cubic f (as in this work), the value of α_2 given by Eqns. (B-4) - (B-6), may not be the true minimum along $\hat{d}_k$; however, as stated in Chapter V for the modified D-F-P method, α_2 need only be accurate enough to insure that $\underline{p}^T \underline{q} > 0$. A logical choice for α_0 is zero, since then $f_0 = f(\underline{x}_k)$, a value known from the previous

---

[1] From Luenberger [23 p. 142].

iteration. If α_1 can be chosen such that it and α_0 "bracket" α^*, the true minimum, is greatly increased. A clever way of achieving this[2] (if f is not varying rapidly) is to make α_1 depend upon f_0 and f_0' in the following manner:

$$\alpha_1 = \min [-2f_0/f_0', \alpha_{max}], \tag{B-9}$$

where α_{max} is a constant, limiting value. The expression $-2f_0/f_0'$ is seen in Figure B.1 to represent the distance from $\alpha_0 = 0$ to the point of intersection of the tangent to f at α_0 and the α axis. Taking twice that distance for α_1 gives a point on the other side of the true minimum α^*. The resulting cubic fit is shown as the dotted curve, with α_2 as its minimum. In cases where α_2 is not sufficiently close to α^* to give $\underline{p}^T\underline{g} > 0$, the cubic fit is repeated, with $\alpha_0 = \alpha_2$, and α_1 computed as before. The convergence is quite rapid, generally requiring no more than two fits.

Finally, it should be noted that the use of $\hat{\underline{d}}_k$ rather than $\underline{d}_k$ in Eqn. (B-8) means that $\Delta\alpha$ is the magnitude of the increment ($\Delta\alpha\hat{\underline{d}}_k$), which should be small to obtain accurate derivatives. Use of $\underline{d}_k$, whose norm varies between iterations, would give unpredictable results.

---

[2] Fletcher and Powell [34] first proposed this scheme.

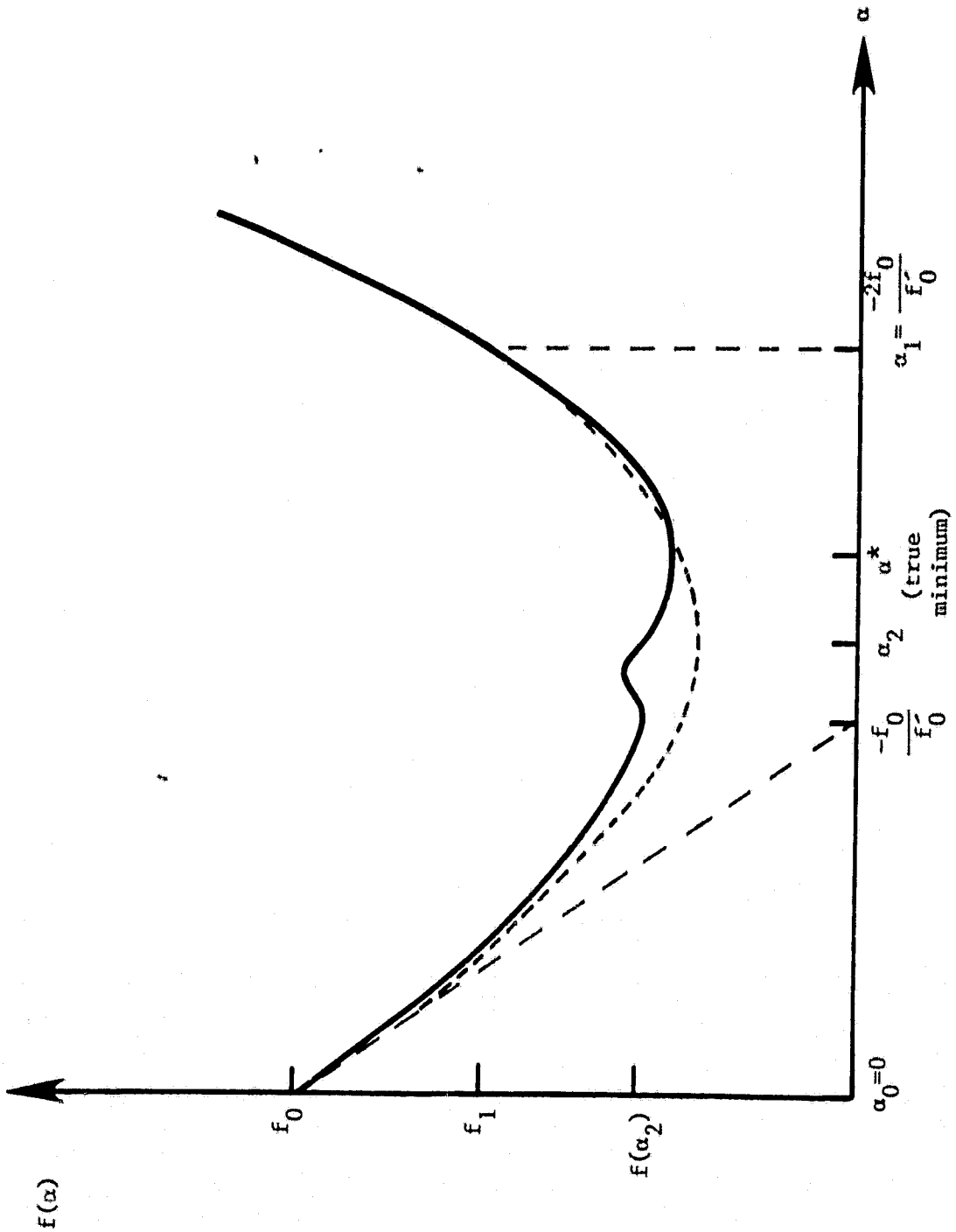


Figure B.1 Line Search by Cubic Fit

APPENDIX C

DETERMINATION OF TANGENT POINTS AND CENTER OF CIRCLE AT A CORNER POINT

In order to evaluate some of the constraint equations in Chapter IV, and for purposes of sampling an aircraft's position as described in Chapter VII, it is necessary to know the cartesian coordinates of the tangent points on a trajectory (see Figure C.1). The value of R_{min} is determined by the dynamic and passenger comfort constraints (Eqn. 4-13), and the corner point coordinates are the unknowns that the optimization scheme determines.

Ground Plane

First, the value of α is given (by the Law of Cosines) as:

$$\alpha = \cos^{-1} \left[\frac{a^2 + b^2 - c^2}{2ab} \right] \qquad \text{(C-1)}$$

where

$$a = \overline{P_1P_2},\ b = \overline{P_2P_3},\ \text{and}\ c = \overline{P_1P_3}.$$

The distance l from the corner point of interest (x_2,y_2) is

$$l = \frac{R_{min}}{\tan\frac{\alpha}{2}}, \qquad \text{(C-2)}$$

so that the tangent points may then be expressed as

$$x_{t1} = \frac{l}{\overline{P_1P_2}} (x_1 - x_2) + x_2 \qquad \text{(C-3a)}$$

$$y_{t1} = \frac{1}{\overline{P_1P_2}} (y_1 - y_2) + y_2 \qquad \text{(C-3b)}$$

$$x_{t2} = \frac{1}{\overline{P_2P_3}} (x_3 - x_2) + x_2 \qquad \text{(C-3c)}$$

$$y_{t2} = \frac{1}{\overline{P_2P_3}} (y_3 - y_2) + y_2, \qquad \text{(c-3d)}$$

where x_i, y_i are the x and y coordinates of the i-th corner point.

The linearity of the segments requires that

$$\frac{z_{t1} - z_2}{z_1 - z_2} = \frac{1}{\overline{P_1P_2}} \qquad \text{(C-4a)}$$

$$\frac{z_{t2} - z_2}{z_3 - z_2} = \frac{1}{\overline{P_2P_3}}, \qquad \text{(C-4b)}$$

from which the values of z_{t1} and z_{t2} may be obtained (z_i is the z coordinate of the i-th corner point).

Although the coordinates of the tangent points in the ground plane and R_{min} specify the circular arc, it is more convenient to express it in terms of R_{min} and the coordinates of the circle's center, (x_c, y_c). This point is just the place of intersection of the two radii shown in Figure C.1. The coordinates are given by

$$x_c = \frac{b_2 - b_1}{m_1 - m_2} \qquad \text{(C-5a)}$$

$$y_c = \frac{m_1 b_2 - m_2 b_1}{m_1 - m_2} \qquad \text{(C-5b)}$$

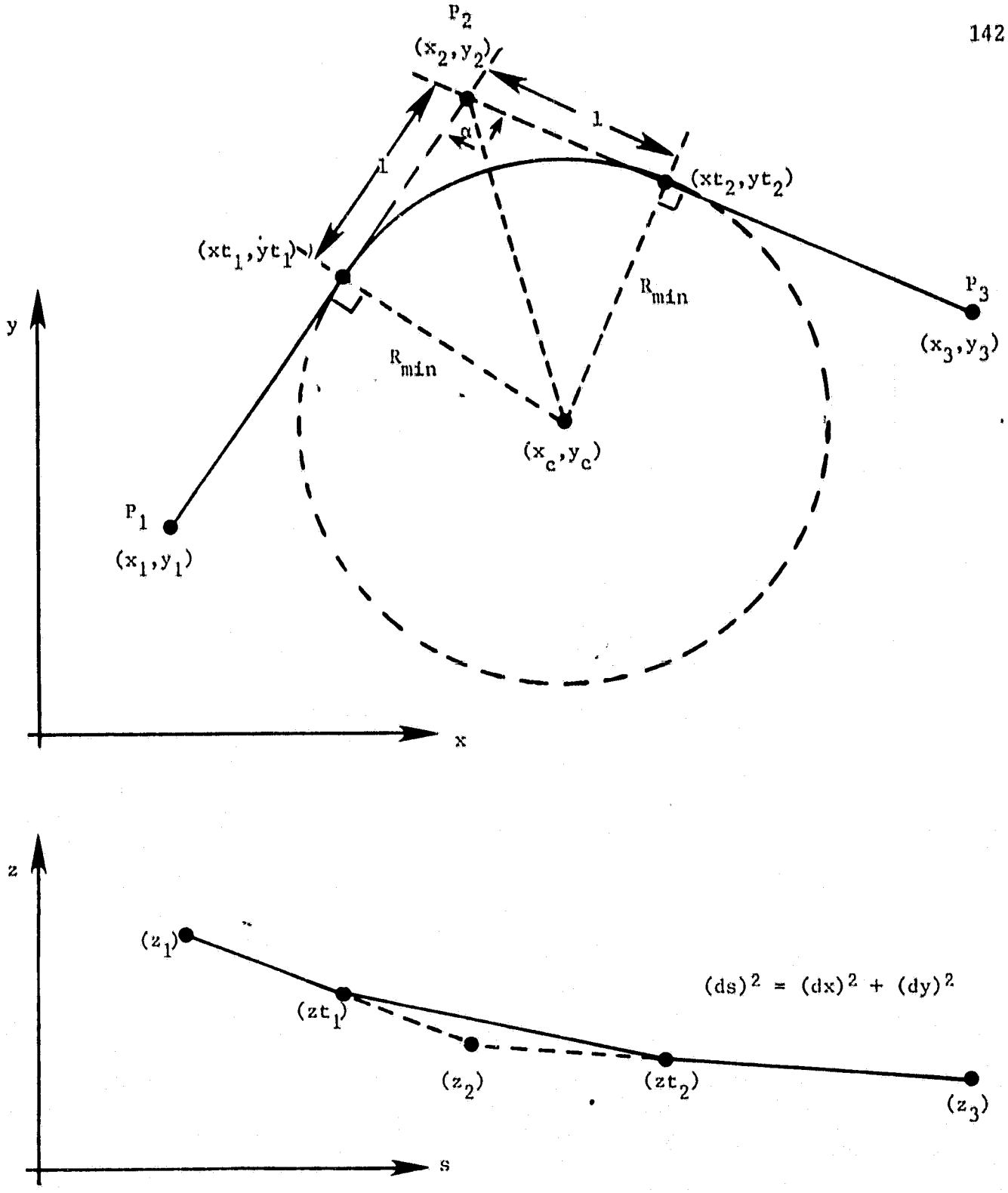


Figure C.1 The Tangent Points

where

$$M_1 = \frac{x_1 - x_{t1}}{y_{t1} - y_1} \tag{C-5c}$$

$$m_2 = \frac{x_3 - x_{t2}}{y_{t2} - y_3} \tag{C-5d}$$

$$b_1 = y_{t1} - m_1 x_{t1} \tag{C-5e}$$

$$b_2 = y_{t2} - m_2 X_{t2} \tag{C-5f}$$

APPENDIX D

MANIP2 AND CYCLIC PROGRAM LISTINGS

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE7,TAPE6,TAPE8,TAPE9,TAPMAI   10
     $E10)                                                                MAI   20
C ..................................................................... MAI   30
C .                                                                   . MAI   40
C .                    * * * *  M A N I P 2  * * * *                  . MAI   50
C .                                                                   . MAI   60
C .                 MINIMAL AIRCRAFT NOISE IMPACT PROGRAM             . MAI   70
C .                                                                   . MAI   80
C .                                                                   . MAI   90
C .               AUTHORS:  R. HELTON AND R. CHANG                    . MAI  100
C .                         UNIVERSITY OF VIRGINIA                    . MAI  110
C .                         CHARLOTTESVILLE, VA                       . MAI  120
C .                                                                   . MAI  130
C .               REVISED:  FEB. 3, 1981                              . MAI  140
C .                                                                   . MAI  150
C ..................................................................... MAI  160
C .                                                                   . MAI  170
C .     THIS PROGRAM USES NONLINEAR OPTIMIZATION TECHNIQUES TO        . MAI  180
C .     FIND THE SET OF AIRCRAFT TRAJECTORIES WITH THE MINIMUM        . MAI  190
C .     NOISE IMPACT INDEX (NII) FOR A SPECIFIED MIX OF AIRCRAFT      . MAI  200
C .     AND FOR THE PARTICULAR POPULATION DISTRIBUTION INPUT.         . MAI  210
C .                                                                   . MAI  220
C .     THE SUBROUTINES USED ARE:                                     . MAI  230
C .                                                                   . MAI  240
C .        JPARAMS (NMACFTN PACKAGE ON NASA LRC PUBLIC LIB)           . MAI  250
C .        COST,COST1,AACCS,MONIT,RESULT,CRCSCVR,ONEVTON,MPLY,TRNSPCS,. MAI  260
C .        FGRAD,GAFA,EFFCP,LINESCH,CHECK,CORNER,DYNARID,NII,NOISEL   . MAI  270
C .                                                                   . MAI  280
C ..................................................................... MAI  290
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITMAI  300
     $(3,200,3),XO(3),XF(3),YO(3),YF(3),ZO(3),ZF(3),NPOSITS(3),RHO(576),MAI  310
     $ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,FATOFC,FATOFW          MAI  320
      COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,ZWEIT1,GMAMIN,GMAMAX,NSAMP   MAI  330
      COMMON /THRESH/ ALMAX,MAXFLIT,TWEIT1,TWEIT2,NPLANE                 MAI  340
      COMMON /CROSS/ XBEGIN,XFINAL,YDIS,GWEIT1,GWEIT2                    MAI  350
      COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,5),SYT1(3,5),SZT1(MAI  360
     $3,5),SXT2(3,5),SYT2(3,5),SZT2(3,5),SANGLE(3,3)                     MAI  370
      COMMON /SAMPLE/ SAMPT,PERIOD,SAMLTH,NAC(3,32),CNOIS(16,2)          MAI  380
      DIMENSION XNOW(30,1), DELTAX(30), LABEL(8)                         MAI  390
      DIMENSION XPORT(2), YPORT(3), JN(30)                               MAI  400
      DIMENSION NDC830(3,2), NDC9SAM(3,2), NDC10(3,2), N707SAM(3,2), N72MAI  410
     $0(3,2), N727200(3,2), N727SAM(3,2), N737100(3,2), N737SAM(3,2), N7MAI  420
     $47200(3,2), NL1011(3,2), NA300(3,2), NBAC111(3,2), NVC10(3,2), NCVMAI  430
     $990(3,2), NSST(3,2), NAMAC(32)                                     MAI  440
      EQUIVALENCE (NAC(1),NDC830(1)), (NAC(7),NDC9SAM(1)), (NAC(13),NDC1MAI  450
     $0(1)), (NAC(19),N707SAM(1)), (NAC(25),N72C(1)), (NAC(31),N727200(1MAI  460
     $)), (NAC(37),N727SAM(1)), (NAC(43),N737100(1)), (NAC(49),N737SAM(1MAI  470
     $)), (NAC(55),N747200(1)), (NAC(61),NL1011(1)), (NAC(67),NA300(1)),MAI  480
     $ (NAC(73),NBAC111(1)), (NAC(79),NVC10(1)), (NAC(85),NCV990(1)), (NMAI  490
     $AC(91),NSST(1))                                                    MAI  500
      DATA ((CNOIS(I,J),J=1,2),I=1,16)/169.74,28.32,154.28,24.65,158.22,MAI  510
     $26.80,147.11,23.60,152.68,21.95,151.47,22.56,136.28,18.74,166.74,2MAI  520
     $9.52,159.31,26.96,151.89,24.78,148.94,24.49,178.47,37.20,162.72,27MAI  530
     $.06,154.65,22.95,169.34,27.16,148.22,15.96/                        MAI  540
      DATA NAMAC/"DC-8-30   ","          ","DC-9 W/SAM"," ENG       ","DCMAI  550
     $-10-10  ","          ","707 W/SAM ","ENG       ","720       ","   MAI  560
     $       ","727-200   ","          ","727 W/SAM ","ENG       ","737-MAI  570
     $100/20","0          ","737 W/SAM ","ENG       ","747-200   ","     MAI  580
     $     ","L1011     ","          ","A300      ","          ","BAC-11MAI  590
     $1   ","          ","VC-10     ","          ","CV-990    ","       MAI  600
     $   ","CONCORDE SN","ST       "/                                    MAI  610
      DATA NAC/48C*0/                                                    MAI  620
      NAMELIST /ACMIX/ NDC830,NDC9SAM,NDC10,N707SAM,N72C,N727200,N727SAMMAI  630
     $,N737100,N737SAM,N747200,NL1011,NA300,NBAC111,NVC10,NCV990,NSST    MAI  640
```

```
C                                                                       MAI  650
C .....................................................................  MAI  660
C .                                                                     .  MAI  670
C .   INPUT PARAMETERS:                                                .  MAI  680
C .   IENTRY = "FIXED" FOR FIXED ENTRY PTS.                            .  MAI  690
C .         = "VARIAB" FOR VARIABLE ENTRY PTS.                          .  MAI  700
C .   ICNTINU = 0, INITIAL TRAJ'S ARE STRAIGHT LINES                   .  MAI  710
C .           1, INITIAL TRAJ'S ARE USER INPUT                          .  MAI  720
C .   SAMPT = TIME IN SECONDS BETWEEN NOISE LEVEL SAMPLES               .  MAI  730
C .   PERIOD = TIME IN HRS. OVER WHICH THE SIMULATION OCCURS            .  MAI  740
C .   ACSPD = A/C SPEED IN METERS/SEC.                                  .  MAI  750
C .   NTRAJ = NUMBER OF FLIGHT TRAJECTORIES                             .  MAI  760
C .   NMAP = NUMBER OF POPULATION POINTS ON MAP                         .  MAI  770
C .   NSEG = NUMBER OF LINE SEGMENTS ON EACH TRAJECTORY                 .  MAI  780
C .   MAXIT = MAXIMUM NO. OF ITERATIONS ALLOWED                         .  MAI  790
C .   IF ONLY RESULT OF INITIAL CONDITION IS NEEDED, SET MAXIT = 0      .  MAI  800
C .   DELH = PERTURBATION (METERS) IN X,Y DIRECTIONS AT CORNER          .  MAI  810
C .   POINTS FOR GRADIENT CALCULATION                                   .  MAI  820
C .   STOPCHG = STOP CRITERION FOR SUCCESSIVE COST CHANGE               .  MAI  830
C .   XO,YO,ZO,XF,YF,ZF = STARTING AND FINAL POINTS OF TRAJECTORIES     .  MAI  840
C .   XPORT,YPORT = AIRPORT LOCATION (METERS)                           .  MAI  850
C .   XM,YM,ZM (I,J) = COORD'S OF J-TH CORNER PT. ON I-TH TRAJ.         .  MAI  860
C .   WEIT1 = PENALTY WEIGHT ON DYN. CONSTR. ON TRAJ'S                  .  MAI  870
C .   WEIT2 = PENALTY WEIGHT ON CONSTR. ON FINAL ANGLE                  .  MAI  880
C .   RADIUS = MIN ALLOWED TURNING RADIUS (METERS)                      .  MAI  890
C .   FATOFW = FUEL AND TIME OF FLIGHT WEIGHT                           .  MAI  900
C .   XMIN,XMAX,YMIN,YMAX = BOUNDARIES OF AREA EXCLUDED                 .  MAI  910
C .                         FROM MAX. NOISE (THRESHOLD) CONSTR.         .  MAI  920
C .   ALMAX = MAX. ALLOWED A-LEVEL NOISE                                .  MAI  930
C .   RATIO = FRACTION OF FLIGHTS ALLOWED TO VIOLATE MAX. A-LEVEL       .  MAI  940
C .   TWEIT1 = PENALTY WEIGHT ON MAX. NOISE AND NUMBER CONSTR.          .  MAI  950
C .   TWEIT2 = NOT USED NOW, BUT SOME VALUE MUST BE INPUT               .  MAI  960
C .   XBEGIN,XFINAL = BOUNDARIES ON X-AXIS WHERE SEPARATING             .  MAI  970
C .                   CONSTR. IS IMPOSED                                .  MAI  980
C .   YDIS = MIN. SEPARATING DISTANCE (Y-AXIS) BETWEEN TRAJ'S.          .  MAI  990
C .   CWEIT1,CWEIT2 = PENALTY WEIGHTS ON SEP. AND CROSSOVER             .  MAI 1000
C .                   CONSTR'S. RESPECTIVELY                            .  MAI 1010
C .   IPROF = 0, NO PROFILE OPTIMIZATION                                .  MAI 1020
C .           1, PROFILE OPTIMIZATION OCCURS                            .  MAI 1030
C .   NOTE:  IPROF IS ALWAYS SET = 1 IN THIS PROGRAM                    .  MAI 1040
C .   ZWEIT1 = PENALTY WEIGHT ON VERTICAL SLOPE CONSTR.                 .  MAI 1050
C .   GMAMAX = MAX. ALLOWED VERTICAL ANGLE ON EACH SEG.                 .  MAI 1060
C .   GMAMIN = MIN. ALLOWED VERTICAL ANGLE ON EACH SEG.                 .  MAI 1070
C .   PENCRIT = MIN. SIGNIFICANT DYN. PENALTY LEVEL                     .  MAI 1080
C .   AWEIT = SCALING ON HII                                            .  MAI 1090
C .                                                                     .  MAI 1100
C .   ACMIX = NAMELIST:  ACTYPE (I,1) = NO. OF ACTYPE ON TRAJ. I       .  MAI 1110
C .                                     IN DAYTIME                      .  MAI 1120
C .                      ACTYPE (I,2) = NO. OF ACTYPE ON TRAJ. I       .  MAI 1130
C .                                     AT NIGHT                        .  MAI 1140
C .....................................................................  MAI 1150
C                                                                       MAI 1160
      CALL DATE (DAT)                                                   MAI 1170
      ENCODE (10,9010,IDAT) DAT                                         MAI 1180
      CALL REMARK (IDAT)                                                MAI 1190
      CALL JPARAMS (JN)                                                 MAI 1200
      ENCODE (7,9020,JOBNAME) JN(1)                                     MAI 1210
      CALL REMARK (JOBNAME)                                             MAI 1220
      WRITE (6,9110)                                                    MAI 1230
      WRITE (6,903C)                                                    MAI 1240
      WRITE (6,9040) JOBNAME,IDAT                                       MAI 1250
      NPLANE = 0                                                        MAI 1260
C                                                                       MAI 1270
C .....................................................................  MAI 1280
```

```
C .                                                                    . MAI 1290
C .   INPUT ALL THE INFORMATION                                        . MAI 1300
C .                                                                    . MAI 1310
C ..................................................................... MAI 1320
C                                                                      MAI 1330
      READ (7,9120) (LABEL(I),I=1,8)                                   MAI 1340
      READ (7,9050) IENTRY                                             MAI 1350
      IF (IENTRY.NE."VARIA".AND.IENTRY.NE."FIXED") CALL ERROR (4,"ILLEGAMAI 1360
     $L ENTRY POINT TYPE, CHECK DATA")                                 MAI 1370
C                                                                      MAI 1380
C ..................................................................... MAI 1390
C .                                                                    . MAI 1400
C .   ISEG1 = 1 : FIXED ENTRY PTS., 1ST SEG. INCLUDED IN ALL CALCS     . MAI 1410
C .   ISEG1 = 2 : VARIABLE ENTRY PTS., 1ST SEG. EXCLUDED               . MAI 1420
C .                                                                    . MAI 1430
C ..................................................................... MAI 1440
C                                                                      MAI 1450
      ISEG1 = 1                                                        MAI 1460
      IF (IENTRY.EQ."VARIA") ISEG1 = 2                                 MAI 1470
      READ (7,*) ICNTINL,SAMPT,PERICD,ACSPD                            MAI 1480
      READ (7,*) NTRAJ,NMAP,NSEG,MAXIT                                 MAI 1490
      READ (7,*) DELH,STOPCHG                                          MAI 1500
      SAMLTH = SAMPT*ACSPD                                             MAI 1510
      DO 30 I = 1,NTRAJ                                                MAI 1520
        READ (7,*) XC(I),YO(I),ZO(I)                                   MAI 1530
        READ (7,*) XF(I),YF(I),ZF(I)                                   MAI 1540
        READ (7,*) XPORT(I),YPORT(I)                                   MAI 1550
C                                                                      MAI 1560
C ..................................................................... MAI 1570
C .                                                                    . MAI 1580
C .   CALCULATE INITIAL CORNER POINTS                                  . MAI 1590
C .                                                                    . MAI 1600
C ..................................................................... MAI 1610
C                                                                      MAI 1620
        XM(I,1) = XC(I)                                                MAI 1630
        YM(I,1) = YO(I)                                                MAI 1640
        ZM(I,1) = ZO(I)                                                MAI 1650
        XM(I,NSEG+1) = XF(I)                                           MAI 1660
        YM(I,NSEG+1) = YF(I)                                           MAI 1670
        ZM(I,NSEG+1) = ZF(I)                                           MAI 1680
        XM(I,NSEG+2) = (XPORT(I)+XF(I))/2.                             MAI 1690
        YM(I,NSEG+2) = (YPORT(I)+YF(I))/2.                             MAI 1700
        ZM(I,NSEG+2) = ZF(I)/2.                                        MAI 1710
        IF (ICNTINL.EQ.1) GO TO 20                                     MAI 1720
        DO 10 J = 2,NSEG                                               MAI 1730
          XM(I,J) = XC(I)+(FLOAT(J-1)/FLOAT(NSEG))*(XF(I)-XO(I))       MAI 1740
          YM(I,J) = YO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(YF(I)-YO(I))       MAI 1750
          ZM(I,J) = ZO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(ZF(I)-ZO(I))       MAI 1760
   10     CONTINUE                                                     MAI 1770
        GO TO 30                                                       MAI 1780
   20   READ (7,*) (XM(I,J),YM(I,J),ZM(I,J),J=2,NSEG)                  MAI 1790
   30   CONTINUE                                                       MAI 1800
      DO 40 I = 1,NMAP                                                 MAI 1810
        DO 40 J = 4,6                                                  MAI 1820
        ARRAY(I,J) = 0.                                                MAI 1830
   40   CONTINUE                                                       MAI 1840
      READ (7,*) PWEIT1,PWEIT2,RADIUS,FATCFW                           MAI 1850
      READ (7,*) XMIN,XMAX,YMIN,YMAX                                   MAI 1860
      READ (7,*) ALMAX,RATIO,TWEIT1,TWEIT2                             MAI 1870
      READ (7,*) XBEGIN,XFINAL,YDIS,CWEIT1,CWEIT2                      MAI 1880
      READ (7,*) IPRCF,ZWEIT1,GMAMIN,GMAMAX,PENCRIT,AWEIT              MAI 1890
      READ (10,*) ((ARRAY(I,J),J=1,3),(ARRAY(I,J),J=6,9),RHO(I),I=1,NMAPMAI 1900
     $)                                                                MAI 1910
      READ (7,ACMIX)                                                   MAI 1920
```

```
      IPROF = 1                                                          MAI 1930
      DO 50 I = 1,NTRAJ                                                  MAI 1940
        DO 50 L = 1,32                                                   MAI 1950
        NPLANE = NPLANE+NAC(I,L)                                         MAI 1960
   50   CONTINUE                                                         MAI 1970
      DO 60 I = 1,NMAP                                                   MAI 1980
        ARRAY(I,10) = 1.                                                 MAI 1990
        IF (XMIN.LE.ARRAY(I,1).AND.ARRAY(I,1).LE.XMAX.AND.YMIN.LE.ARRAY(MAI 2000
     $  I,2).AND.ARRAY(I,2).LE.YMAX.AND.ARRAY(I,3).NE.0.) ARRAY(I,10) = MAI 2010
     $  0.                                                               MAI 2020
   60   CONTINUE                                                         MAI 2030
      MAXFLIT = IFIX(NPLANE*RATIO+0.5)                                   MAI 2040
C                                                                        MAI 2050
C ..................................................................     MAI 2060
C .                                                                .     MAI 2070
C .   PRINT INFORMATION INPUT                                      .     MAI 2080
C .                                                                .     MAI 2090
C ..................................................................     MAI 2100
C                                                                        MAI 2110
      WRITE (6,9130) (LABEL(I),I=1,8)                                    MAI 2120
      WRITE (6,9140) MAXIT,NTRAJ,NSEG,NMAP,DELH,STCPCHG                  MAI 2130
      WRITE (6,9060) SAMPT,PERIOD,ACSPD                                  MAI 2140
      IF (IENTRY.EQ."FIXED") WRITE (6,9070)                              MAI 2150
      IF (IENTRY.EQ."VARIAN") WRITE (6,9080)                             MAI 2160
      IF (IPROF.EQ.1) WRITE (6,9090)                                     MAI 2170
      IF (IPROF.EQ.0) WRITE (6,9100)                                     MAI 2180
      WRITE (6,9150) ALMAX,MAXFLIT,XMIN,XMAX,YMIN,YMAX                   MAI 2190
      WRITE (6,9160) XBEGIN,XFINAL,YDIS                                  MAI 2200
      WRITE (6,9170) RADILS,PWEIT1,PWEIT2,ZWEIT1,TWEIT1,TWEIT2,CWEIT1,CWMAI 2210
     $EIT2,GMAMIN,GMAMAX,PENCRIT,AWEIT,FATOFW                            MAI 2220
      DO 90 I = 1,NTRAJ                                                  MAI 2230
        WRITE (6,9180) I,XO(I),YO(I),ZO(I),XF(I),YF(I),ZF(I)             MAI 2240
        NSEG1 = NSEG+1                                                   MAI 2250
        DO 70 J = 1,NSEG1                                                MAI 2260
          WRITE (6,9200) J,XM(I,J),YM(I,J),ZM(I,J)                       MAI 2270
   70     CONTINUE                                                       MAI 2280
        WRITE (6,9210)                                                   MAI 2290
        DO 80 L = 1,31,2                                                 MAI 2300
          IF (NAC(I,L).NE.0.OR.NAC(I,L+1).NE.0) WRITE (6,9220) NAMAC(L),MAI 2310
     $    NAMAC(L+1),NAC(I,L),NAC(I,L+1)                                 MAI 2320
   80     CONTINUE                                                       MAI 2330
        WRITE (6,9190) XPORT(I),YPORT(I)                                 MAI 2340
   90   CONTINUE                                                        MAI 2350
      N = 2*(NSEG-1)*NTRAJ                                               MAI 2360
      NVAR = 2                                                           MAI 2370
      IF (IPROF.EQ.1) N = 3*(NSEG-1)*NTRAJ                               MAI 2380
      IF (IPROF.EQ.1) NVAR = 3                                           MAI 2390
      DO 100 I = 1,NTRAJ                                                 MAI 2400
        NSEG1 = NSEG-1                                                   MAI 2410
        DO 100 J = 1,NSEG1                                               MAI 2420
        DO 100 K = 1,NVAR                                                MAI 2430
        L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                                MAI 2440
        IF (K.EQ.1) XNOW(L,1) = XM(I,J+1)                                MAI 2450
        IF (K.EQ.2) XNOW(L,1) = YM(I,J+1)                                MAI 2460
        IF (K.EQ.3) XNOW(L,1) = ZM(I,J+1)                                MAI 2470
  100   CONTINUE                                                         MAI 2480
      DO 110 I = 1,N                                                     MAI 2490
        DELTAX(I) = DELH                                                 MAI 2500
  110   CONTINUE                                                         MAI 2510
C                                                                        MAI 2520
C ..................................................................     MAI 2530
C .                                                                .     MAI 2540
C .   START OPTIMIZATION                                           .     MAI 2550
C .                                                                .     MAI 2560
```

```
C ..................................................................... MAI 2570
C                                                                       MAI 2580
      CALL QNEWTON (MAXIT,STOPCHG,N,XNOW,DELTAX,30,PENCRIT)             MAI 2590
      STOP                                                              MAI 2600
C                                                                       MAI 2610
 9010 FORMAT (A10)                                                      MAI 2620
 9020 FORMAT (A7)                                                       MAI 2630
 9030 FORMAT (//,' NO AMBIENT NOISE INCLUDED',//)                       MAI 2640
 9040 FORMAT (1H1,9X,'.............',/,10X,'.',11X,'.',/,10X,'.',2X,A7,2HMAI 2650
     $X,'.',/,10X,'.',11X,'.',/,10X,'.',A10,1X,'.',/,10X,'.',11X,'.',/,1MAI 2660
     $0X,'.............',/)                                             MAI 2670
 9050 FORMAT (A5)                                                       MAI 2680
 9060 FORMAT (5X,'FOR LDN CALCULATION, SAMPLE TIME = ',F4.0,' SEC.',/,5XMAI 2690
     $,'A/C LANDING PATTERN REPEATS EVERY ',F5.1,' HRS.',/,5X,'A/C SPEEDMAI 2700
     $ = ',F6.2,' M/S',///)                                             MAI 2710
 9070 FORMAT (/,4X,'FIXED ENTRY POINTS',/)                              MAI 2720
 9080 FORMAT (/,4X,'VARIABLE ENTRY POINTS',/)                           MAI 2730
 9090 FORMAT (/,4X,'.. PROFILES OPTIMIZED ..',/)                        MAI 2740
 9100 FORMAT (/,4X,'.. ONLY GROUND TRACKS ARE OPTIMIZED ..',/)          MAI 2750
 9110 FORMAT (1H1,10H**********,///)                                    MAI 2760
 9120 FORMAT (8A10)                                                     MAI 2770
 9130 FORMAT (1X,8A10)                                                  MAI 2780
 9140 FORMAT (///,5X,'MAXIMUM ITERATION SET: ',I2,/,5X,'NUMBER OF TRAJ'MAI 2790
     $JECTORIES: ',I2,/,5X,'NUMBER OF SEGMENTS ON EACH TRAJECTORY: ',I1,MAI 2800
     $/,5X,'NUMBER OF POPULATION POINTS ON THE MAP: ',I4,/,5X,'PERTURB TMAI 2810
     $RAJECTORIES IN X AND Y DIRECTIONS ',F10.5,' METERS FOR CALCULATINGMAI 2820
     $ GRADIENTS',//,5X,'EXIT CRITERION FOR QNEWTON: ',1PE16.9,/)       MAI 2830
 9150 FORMAT (5X,'NO BLOCK MAY RECEIVE ',F5.1,' DB (OR HIGHER) A-LEVEL',MAI 2840
     $' NOISE MORE THAN ',I4,' TIMES A DAY',//,3X,'..THIS CONSTRAINT ','MAI 2850
     $EXCLUDES BLOCKS WITH CENTROIDS INSIDE THE RANGES: ',F8.1,' TO ',FEMAI 2860
     $.1,' (X) AND ',F8.1,' TO ',F8.1,' (Y)',//)                        MAI 2870
 9160 FORMAT (5X,40HWITHIN THE AREA OF X-COORDINATES BETWEEN,F8.1,5H ANDMAI 2880
     $ ,F8.1,41H, THE SEPARATING DISTANCE BETWEEN TRAJECT,5HORIES,/,5X,1MAI 2890
     $H SHOULD BE AT LEAST,F8.1,7H METERS,//)                           MAI 2900
 9170 FORMAT (/,6X,'OTHER CONSTRAINT PARAMETERS',/,4X,'MINIMUM RADIUS','MAI 2910
     $ OF CURVATURE (IN X-Y PLANE PROJECTION) = ',1PE9.2,/,4X,'DYNAMIC PMAI 2920
     $ENALTY WEIGHTS:  PWEIT1 = ',E9.2,'  PWEIT2 = ',E9.2,'  ZWEIT1 = ',MAI 2930
     $E9.2,/,4X,'MAXIMUM NOISE LEVEL (THRESHOLD) ','PENALTY WEIGHTS:  TWMAI 2940
     $EIT1 = ',E9.2,'TWEIT2 = ',E9.2,/,4X,'SEPARATION PENALTY WEIGHTS:  MAI 2950
     $CWEIT1 = ',E9.2,'  CWEIT2 = ',E9.2,/,4X,'GMAMIN = ',OPF5.1,' GMAMAXMAI 2960
     $X = ',F5.1,'PENCRIT = ',1PE9.2,' AWEIT = ',E9.2,5X,'FATOFW = ',E9.MAI 2970
     $2,///,5X,'INFORMATION OF EACH INITIAL TRAJECTORY:',//)            MAI 2980
 9180 FORMAT (10X,15HFLIGHTPATH NO: ,I1,/,12X,13HINITIAL X,Y,Z,1X,23HCOORMAI 2990
     $DINATES IN METERS: ,3(F8.1,3X),/,12X,11HFINAL X,Y,Z,1X,23HCOORDINMAI 3000
     $ATES IN METERS: ,3(F8.1,3X),/,12X,14HINITIAL CORNER,18H POINT POSIMAI 3010
     $TIONS: ,/,14X,10HCORNER NO.,6X,1HX,9X,1HY,9X,1HZ,/)               MAI 3020
 9190 FORMAT (10X,'RUNWAY LOCATION: ',F8.1,10X,F8.1,//)                 MAI 3030
 9200 FORMAT (18X,I1,6X,F8.1,3X,F8.1,3X,F8.1)                           MAI 3040
 9210 FORMAT (15X,'A/C MIX',T45,'NO. OF A/C',//,T40,'DAYTIME',3X,'NIGHTTMAI 3050
     $IME',//)                                                          MAI 3060
 9220 FORMAT (T20,2A10,T42,I3,T52,I3/)                                  MAI 3070
      END                                                               MAI 3080



      SUBROUTINE COST (IGRAD,TOTAL,ANII,PNALTY,CLOSE,THRESH)            CCS   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITCCS   20
     $(3,200,3),XO(3),XF(3),YO(3),YF(3),ZO(3),ZF(3),NPOSITS(3),RHO(576),CCS   30
     $ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,FATOFC,FATOFW         CCS   40
      COMMON /CORNER/ ANGLE(3,3),POSIT(3,200,3),XCENTP(3,3),YCENTR(3,3),CCS   50
     $ZCENTR(3,3),XT1(3,5),YT1(3,5),ZT1(3,5),XT2(3,5),YT2(3,5),ZT2(3,5),CCS   60
     $NPOSIT(3),DIS(3)                                                  CCS   70
      COMMON /PRINT/ SXCENTP(3,3),SYCENTR(3,3),SXT1(3,5),SYT1(3,5),SZT1(CCS   80
```

```
     13,5),SXT2(3,5),SYT2(3,5),SZT2(3,5),SANGLE(3,2)                    CCS   60
      EATOFC=0.                                                         CCS  100
      TOTAL=FATOFC                                                      CCS  110
      ANII=TOTAL                                                        CCS  120
      CLOSE=ANII                                                        CCS  130
      THRESH=CLOSE                                                      CCS  140
      PNALTY=THRESH                                                     CCS  150
      DO 10 I = 1,NMAP                                                  CCS  160
        DO 10 J = 4,5                                                   CCS  170
        ARRAY(I,J) = 0.                                                 CCS  180
10    CONTINUE                                                          CCS  190
      CALL DWNARID (PNALTY)                                             CCS  200
      IF (PNALTY.LT.PENCRIT) GO TO 20                                   CCS  210
      TOTAL = PNALTY                                                    CCS  220
      IF (MAXIT.NE.0) RETURN                                            CCS  230
20 CALL NOISEL (THRESH)                                                 CCS  240
      CALL NII (ANII)                                                   CCS  250
      IF (NTRAJ.EQ.1) GO TO 30                                          CCS  260
      CALL CROSCVR (NPOSIT,POSIT,CLOSE)                                 CCS  270
30 TOTAL = ANII+THRESH+CLOSE+PNALTY+FATOFC                              CCS  280
      IF (IGRAD.EQ.1) RETURN                                            CCS  290
      DO 60 I = 1,NTRAJ                                                 CCS  300
        SXT1(I,1) = XT1(I,1)                                            CCS  310
        SXT2(I,1) = XT2(I,1)                                            CCS  320
        SYT1(I,1) = YT1(I,1)                                            CCS  330
        SYT2(I,1) = YT2(I,1)                                            CCS  340
        SZT1(I,1) = ZT1(I,1)                                            CCS  350
        SZT2(I,1) = ZT2(I,1)                                            CCS  360
        DO 40 J = 2,NSEG                                                CCS  370
          SXCENTR(I,J-1) = XCENTR(I,J-1)                                CCS  380
          SYCENTR(I,J-1) = YCENTR(I,J-1)                                CCS  390
          SXT1(I,J) = XT1(I,J)                                          CCS  400
          SYT1(I,J) = YT1(I,J)                                          CCS  410
          SXT2(I,J) = XT2(I,J)                                          CCS  420
          SYT2(I,J) = YT2(I,J)                                          CCS  430
          SZT1(I,J) = ZT1(I,J)                                          CCS  440
          SZT2(I,J) = ZT2(I,J)                                          CCS  450
          SANGLE(I,J-1) = ANGLE(I,J-1)                                  CCS  460
40        CONTINUE                                                      CCS  470
        NPOSITS(1) = NPOSIT(I)                                          CCS  480
        IPOST = NPOSITS(I)                                              CCS  490
        DO 50 J = 1,IPOST                                               CCS  500
          SPOSIT(I,J,1) = POSIT(I,J,1)                                  CCS  510
          SPOSIT(I,J,2) = POSIT(I,J,2)                                  CCS  520
          SPOSIT(I,J,3) = POSIT(I,J,3)                                  CCS  530
50        CONTINUE                                                      CCS  540
60    CONTINUE                                                          CCS  550
      RETURN                                                            CCS  560
      END                                                               CCS  570


      SUBROUTINE COST1 (IGRAD,N,F,X,ANII,PNALTY,CLOSE,THRESH,NDEC)      CO1   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITCO1   20
     $(3,200,3),XC(3),XF(3),YC(3),YF(3),ZC(3),ZF(3),NPOSITS(3),RHO(576),CO1   30
     $ISEG1,IPROF,MAXIT,PENCRIT,ANEIT,TVICLA,ALWP,FATOFC,FATOFW         CO1   40
      DIMENSION X(NDEC,1)                                               CO1   50
      NVAR = 2                                                          CO1   60
      IF (IPROF.EQ.1) NVAR = 3                                          CO1   70
      DO 10 I = 1,NTRAJ                                                 CO1   80
        NSEG1 = NSEG-1                                                  CO1   90
        DO 10 J = 1,NSEG1                                               CO1  100
        DO 10 K = 1,NVAR                                                CO1  110
```

```
      L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                                 CC1  120
      IF (K.EQ.1) XM(I,J+1) = X(L,1)                                    CC1  130
      IF (K.EQ.2) YM(I,J+1) = X(L,1)                                    CC1  140
      IF (K.EQ.3) ZM(I,J+1) = X(L,1)                                    CC1  150
   10 CONTINUE                                                          CC1  160
      CALL COST (IGRAD,F,ANII,PNALTY,CLOSE,THRESH)                      CC1  170
      RETURN                                                            CC1  180
      END                                                               CC1  190


      FUNCTION AACOS (A,B,C)                                            AAC   10
      X = (A**2+B**2-C**2)/(2.*A*B)                                     AAC   20
      IF (ABS(X).LT.1.1) GO TO 10                                       AAC   30
      WRITE (6,9010) X,A,B,C                                            AAC   40
      STOP                                                              AAC   50
   10 IF (X.GT.1.) X = 1.                                               AAC   60
      IF (X.LT.-1.) X = -1.                                             AAC   70
      AACOS = ACOS(X)                                                   AAC   80
      RETURN                                                            AAC   90
C                                                                       AAC  100
 9010 FORMAT (29H TROUBLE IN AACOS, X,A,B,C = ,4(1PE16.9,3X))           AAC  110
      END                                                               AAC  120


      SUBROUTINE MONIT (IT,X,N,F,ANII,PNALTY,CLOSE,THRESH,NDEC)         MON   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),APPAY(576,9),SPOSITMON   20
     $(3,200,3),XO(3),YF(3),YC(3),YF(3),ZO(3),ZF(3),NPCSITS(3),PHC(576),MON   30
     $ISEG1,IPROF,MAXIT,PENCRT,AWEIT,TVIOLA,ALWP,FATOFC,FATOFW          MON   40
      COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,5),SYT1(3,5),SZT1(MON   50
     $3,5),SXT2(3,5),SYT2(3,5),SZT2(3,5),SANGLE(3,3)                    MON   60
      DIMENSION X(NDEC,1)                                               MON   70
      PI = ATAN(1.)*4.                                                  MON   80
      NVAR = 2                                                          MON   90
      IF (IPROF.EQ.1) NVAR = 3                                          MON  100
      DO 10 I = 1,NTRAJ                                                 MON  110
        NSEG1 = NSEG-1                                                  MON  120
        DO 10 J = 1,NSEG1                                               MON  130
        DO 10 K = 1,NVAR                                                MON  140
        L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                               MON  150
        IF (K.EQ.1) XM(I,J+1) = X(L,1)                                  MON  160
        IF (K.EQ.2) YM(I,J+1) = X(L,1)                                  MON  170
        IF (K.EQ.3) ZM(I,J+1) = X(L,1)                                  MON  180
   10   CONTINUE                                                        MON  190
      WRITE (6,80) IT,F,ANII,PNALTY,CLOSE,THRESH,TVIOLA,ALWP,FATOFC     MON  200
      DO 30 I = 1,NTRAJ                                                 MON  210
        WRITE (6,40) I,XM(I,1),YM(I,1),ZM(I,1)                          MON  220
        DO 20 J = 2,NSEG                                                MON  230
          WRITE (6,50) XM(I,J),YM(I,J),ZM(I,J),SXCENTR(I,J-1),SYCENTR(I,MON  240
     $    J-1),SXT1(I,J),SYT1(I,J),SZT1(I,J),SXT2(I,J),SYT2(I,J),SZT2(I,MON  250
     $    J),SANGLE(I,J-1)*180./PI                                     MON  260
   20   CONTINUE                                                        MON  270
        WRITE (6,60) XM(I,NSEG+1),YM(I,NSEG+1),ZM(I,NSEG+1)             MON  280
        NPCSITI = NPCSITS(I)                                            MON  290
        WRITE (8,70) ((SPCSIT(I,J,K),K=1,3),J=1,NPCSITI)                MON  300
   30 CONTINUE                                                          MON  310
      RETURN                                                            MON  320
C                                                                       MON  330
C                                                                       MON  340
   40 FORMAT (12X,"FLIGHT PATH NO: ",I1,/,T20,"CORNER PTS.",T40,"CENTER MON  350
     $OF CIRCLE",T74,"TANGENTIAL PTS.",T110,"ANGLE (DEGREE)",/,14X,"(",FMON  360
```

```
     $7.0,",",F7.0,",",F7.0,",")")                                      MCN  370
  5C FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,",",F7.0,")",",("F7.0,",",F7.0,",",2MCN  380
     $("(",F7.0,",",F7.0,",",F7.0,")"),3X,F6.1)                         MCN  390
  6C FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")")                        MCN  400
  7C FORMAT (3(F8.1,2X))                                                MCN  410
     END                                                                MCN  420



     SUBROUTINE RESLLT (IT,X,N,F,ANII,PNALTY,CLOSE,THRESH,NDEC)         PES   10
     COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITPES   20
    $(3,200,3),XO(3),XF(3),YO(3),YF(3),ZO(3),ZF(3),NPOSITS(3),RHO(576),PES   30
    $ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,FATOFC,FATOFV          PES   40
     COMMON /PRINT/ SXCENTR(3,3),SYCENTR(3,3),SXT1(3,5),SYT1(3,5),SZT1(PES   50
    $3,5),SXT2(3,5),SYT2(3,5),SZT2(3,5),SANGLE(3,3)                     PES   60
     DIMENSION X(NDEC,1)                                                PES   70
     PI = ATAN(1.)*4.                                                   PES   80
     NVAR = 2                                                           PES   90
     IF (IPROF.EQ.1) NVAR = 3                                           PES  100
     DO 10 I = 1,NTRAJ                                                  PES  110
       NSEG1 = NSEG-1                                                   PES  120
       DO 10 J = 1,NSEG1                                                PES  130
       DO 10 K = 1,NVAR                                                 PES  140
       L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                                PES  150
       IF (K.EQ.1) XM(I,J+1) = X(L,1)                                   PES  160
       IF (K.EQ.2) YM(I,J+1) = X(L,1)                                   PES  170
       IF (K.EQ.3) ZM(I,J+1) = X(L,1)                                   PES  180
  10   CONTINUE                                                         PES  190
     WRITE (6,9010) IT,F,ANII,PNALTY,CLOSE,THRESH,TVICLA,ALWP,FATOFC    PES  200
     DO 30 I = 1,NTRAJ                                                  PES  210
       WRITE (6,9020) I,XM(I,1),YM(I,1),ZM(I,1)                         PES  220
       DO 20 J = 2,NSEG                                                 PES  230
         WRITE (6,9030) XM(I,J),YM(I,J),ZM(I,J),SXCENTR(I,J-1),SYCENTR(PES  240
    $     I,J-1),SXT1(I,J),SYT1(I,J),SZT1(I,J),SXT2(I,J),SYT2(I,J),SZT2(PES  250
    $     I,J),SANGLE(I,J-1)*180./PI                                   PES  260
  20     CONTINUE                                                       PES  270
       WRITE (6,9040) XM(I,NSEG+1),YM(I,NSEG+1),ZM(I,NSEG+1)            PES  280
       WRITE (6,9050)                                                   PES  290
       NPOSITI = NPOSITS(I)                                             PES  300
       WRITE (6,9060) ((SPOSIT(I,J,K),K=1,3),J=1,NPOSITI)               PES  310
       WRITE (8,9070) ((SPOSIT(I,J,K),K=1,3),J=1,NPOSITI)               PES  320
  30   CONTINUE                                                         PES  330
     WRITE (9,9080) ((ARRAY(I,J),J=4,5),I=1,NMAP)                       PES  340
     RETURN                                                             PES  350
C                                                                       PES  360
 901C FORMAT (1H1,9X,11HITERATION: ,I2,/,12X,12HTOTAL COST: ,1PE16.9,/,1PES  370
    $2X,17HANNOYANCE (NII): ,1PE16.9,/,12X,31HPENALTY ON DYNAMIC CONSTRPES  380
    $AINT: ,1PE16.9,/,12X,34HPENALTY ON SEPARATING CONSTRAINT: ,1PE16.9PES  390
    $,/,12X,28HPENALTY ON THRESHOLD NOISE: ,1PE16.9,/,12X,"THRESH VICLAPES  400
    $TION = ",E16.9,/,12X,"LWP = ",E16.9,/,12X,"FATOFC = ",E16.9,/)     PES  410
 902C FORMAT (12X,"FLIGHT PATH NO: ",I1,/,T20,"CORNER PTS.",T40,"CENTER PES  420
    $OF CIRCLE",T74,"TANGENTIAL PTS.",T110,"ANGLE (DEGREE)",/,14X,"(",FPES  430
    $7.0,",",F7.0,",",F7.0,")")                                         PES  440
 903C FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")","(",F7.0,",",F7.0,")",2PES  450
    $("(",F7.0,",",F7.0,",",F7.0,")"),3X,F6.1)                          PES  460
 9040 FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")")                       PES  470
 905C FORMAT (//,12X,36HFLIGHT TRAJECTORY,X, Y, Z, IN METERS,/,19X,1HX,1PES  480
    $4X,1HY,14X,1HZ,/)                                                  PES  490
 9060 FORMAT (14X,1PE10.3,5X,E10.3,5X,E10.3,5X)                         PES  500
 9070 FORMAT (3(F8.1,2X))                                               PES  510
 908C FORMAT (2(1PE10.3,1X))                                            PES  520
     END                                                                PES  530
```

```
      SUBROUTINE CROSOVR (NPOSIT,POSIT,FX)                                  CPS   10
C                                                                           CPS   20
C ***********************************************************************  CPS   30
C *                                                                     *   CPS   40
C *   THIS SUBROUTINE CHECKS THE SEPARATION BETWEEN TRAJECTORY          *   CPS   50
C *   PAIRS.  IT IS INTENDED TO WORK ONLY FOR CASES WHERE ALL           *   CPS   60
C *   OF THE TRAJECTORIES SHARE A SEGMENT OF THE X-AXIS.                *   CPS   70
C *                                                                     *   CPS   80
C ***********************************************************************  CPS   90
C                                                                           CPS  100
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITCPS  110
     $(3,200,3),XO(3),XF(3),YO(3),YF(3),ZO(3),ZF(3),NPOSITS(3),RHO(576),CPS  120
     $ISEG1,IPROF,MAXIT,PENCRIT,ABEIT,TVICLA,ALWP,FATOFC,FATOFW             CPS  130
      COMMON /CROSS/ XBEGIN,XFINAL,YOIS,CWEIT1,CWEIT2                       CPS  140
      DIMENSION CROSS(7,10), POSIT(3,200,3), NPOSIT(3)                      CPS  150
      DIMENSION CPOSIT(3,200,3), NCPOS(3), NOTRAJ(3)                        CPS  160
      YINTRP(X,X1,Y1,X2,Y2,X3,Y3,X4,Y4) = (X-X2)*(X-X3)*(X-X4)*Y1/(Y1-X2CPS  170
     $)/(X1-X3)/(X1-X4)+(X-X1)*(X-X3)*(X-X4)*Y2/(X2-X1)/(X2-X3)/(X2-X4)+CPS  180
     $(X-X1)*(X-X2)*(X-X4)*Y3/(X3-X1)/(X3-X2)/(X3-X4)+(X-X1)*(X-X2)*(X-XCPS  190
     $3)*Y4/(X4-X1)/(X4-X2)/(X4-X3)                                         CPS  200
      FX = 0.                                                               CPS  210
      DO 20 I = 1,NTRAJ                                                     CPS  220
        NOTRAJ(I) = 1                                                       CPS  230
        JJ = NPOSIT(I)                                                      CPS  240
        DO 10 J = 1,JJ                                                      CPS  250
          IF ((XBEGIN-POSIT(I,J,1))*(POSIT(I,J,1)-XFINAL).GT.0.) GO TO 20CPS  260
     $    C                                                                CPS  270
   10     CONTINUE                                                         CPS  280
        NOTRAJ(I) = 0                                                       CPS  290
   20   CONTINUE                                                           CPS  300
      II = 0                                                                CPS  310
      DO 40 I = 1,NTRAJ                                                     CPS  320
        IF (NOTRAJ(I).EQ.0) GO TO 40                                        CPS  330
        JJ = NPOSIT(I)                                                      CPS  340
        II = II+1                                                           CPS  350
        NCPOS(II) = JJ                                                      CPS  360
        DO 30 J = 1,JJ                                                      CPS  370
          DO 30 K = 1,3                                                     CPS  380
          CPOSIT(II,J,K) = POSIT(I,J,K)                                     CPS  390
   30     CONTINUE                                                         CPS  400
   40   CONTINUE                                                           CPS  410
      SAMPLE = (XBEGIN-XFINAL)/11.                                          CPS  420
      CROSS(1,1) = XBEGIN-SAMPLE                                            CPS  430
      DO 50 I = 2,10                                                        CPS  440
        CROSS(1,I) = CROSS(1,I-1)-SAMPLE                                    CPS  450
   50   CONTINUE                                                           CPS  460
      DO 110 I = 1,NTRAJ                                                    CPS  470
        DO 100 J = 1,10                                                     CPS  480
          NP1 = NPOSIT(I)-1                                                 CPS  490
          DO 80 K = 1,NP1                                                   CPS  500
            SIGN = (POSIT(I,K,1)-CROSS(1,J))*(POSIT(I,K+1,1)-CROSS(1,J))CPS  510
            IF (SIGN.GE.0.0) GO TO 60                                       CPS  520
            KK = K                                                          CPS  530
            GO TO 90                                                        CPS  540
   60       IF (SIGN.GT.0.0) GO TO 80                                       CPS  550
            IF (POSIT(I,K,1).NE.CROSS(1,J)) GO TO 70                        CPS  560
            CROSS(1+I,J) = POSIT(I,K,2)                                     CPS  570
            GO TO 100                                                       CPS  580
   70       CROSS(1+I,J) = POSIT(I,K+1,2)                                   CPS  590
            GO TO 100                                                       CPS  600
   80       CONTINUE                                                        CPS  610
```

```
   90     IF (KK.EQ.1) CROSS(1+I,J) = YINTRP(CROSS(1,J),POSIT(I,1,1),POSCRS  620
     $ .  IT(I,1,2),POSIT(I,2,1),POSIT(I,2,2),POSIT(I,3,1),POSIT(I,3,2),CRS  630
     $    POSIT(I,4,1),POSIT(I,4,2))                                        CRS  640
          IF (KK.NE.1.AND.KK.NE.NP1) CROSS(1+I,J) = YINTRP(CROSS(1,J),POSCRS  650
     $   /SIT(I,KK-1,1),POSIT(I,KK-1,2),POSIT(I,KK,1),POSIT(I,KK,2),POSICRS  660
     $    T(I,KK+1,1),POSIT(I,KK+1,2),POSIT(I,KK+2,1),POSIT(I,KK+2,2))     CRS  670
          IF (KK.EQ.NP1) CROSS(1+I,J) = YINTRP(CROSS(1,J),POSIT(I,KK-2,1CRS  680
     $    ),POSIT(I,KK-2,2),POSIT(I,KK-1,1),POSIT(I,KK-1,2),POSIT(I,KK,1CRS  690
     $    ),POSIT(I,KK,2),POSIT(I,KK+1,1),POSIT(I,KK+1,2))                 CRS  700
  100     CONTINUE                                                          CRS  710
  110   CONTINUE                                                            CRS  720
C                                                                           CRS  730
C ..........................................................................CRS  740
C .                                                                        .CRS  750
C .   NOW TEST THE NEARNESS OF OR THE CROSSOVER BETWEEN TRAJECTORIES       .CRS  760
C .                                                                        .CRS  770
C .                                                                        .CRS  780
C ..........................................................................CRS  790
C                                                                           CRS  800
      NADD1 = NTRAJ+1                                                       CRS  810
      DO 170 I = 2,NTRAJ                                                    CRS  820
        DO 160 J = 3,NADD1                                                  CRS  830
          IF (I.GE.J) GO TO 160                                             CRS  840
          DO 140 K = 1,10                                                   CRS  850
            DIS1 = CROSS(I,K)-CROSS(J,K)                                    CRS  860
            IF (ABS(DIS1).GE.YDIS) GO TO 120                                CRS  870
            FX = FX+CWEIT1*(YDIS-DIS1)**2                                   CRS  880
  120       IF (K.EQ.10) GO TO 140                                          CRS  890
            DIS2 = CROSS(I,K+1)-CROSS(J,K+1)                                CRS  900
            IF ((DIS1*DIS2).GE.0.) GO TO 140                                CRS  910
            I1 = 1                                                          CRS  920
  130       IF (K+1+I1.GT.10) GO TO 150                                     CRS  930
            DIS3 = CROSS(I,K+1+I1)-CROSS(J,K+1+I1)                          CRS  940
            IF (DIS2*DIS3.LE.0.) GO TO 150                                  CRS  950
            IF (ABS(DIS2).LT.ABS(DIS3)) DIS2 = DIS3                         CRS  960
            I1 = I1+1                                                       CRS  970
            GO TO 130                                                       CRS  980
  140       CONTINUE                                                        CRS  990
            GO TO 160                                                       CRS 1000
  150     FX = FX+CWEIT2*DIS3**2                                            CRS 1010
  160     CONTINUE                                                          CRS 1020
  170   CONTINUE                                                            CRS 1030
      RETURN                                                                CRS 1040
      END                                                                   CRS 1050


      SUBROUTINE ONEWTON (MAXIT,STOPCHG,N,XNOW,DELTAX,NDEC,PENCRIT)        ONE   10
C                                                                           ONE   20
C ..........................................................................ONE   30
C .                                                                        .ONE   40
C .   THIS OPTIMIZATION EMPLOYS SELF-SCALING, RESTARTING,                  .ONE   50
C .   QUASI-NEWTON METHOD.                                                 .ONE   60
C .   REFERENCE:  D.G. LUENBERGER INTRO. TO LINEAR AND NONLINEAR           .ONE   70
C .   PROGRAMMING; P.204, SEC.9.5                                          .ONE   80
C .   MAXIT:  MAXIMUM NUMBER OF ITERATIONS ALLOWED                         .ONE   90
C .   STOPCHG:  STOP IF PERCENTAGE CHANGE IN SUCCESSIVE COSTS IS           .ONE  100
C .   LESS THAN THIS VALUE                                                 .ONE  110
C .   N:  DIMENSION OF THE UNKNOWN X                                       .ONE  120
C .   XNOW:  PRESENT OR INITIAL VALUE OF UNKNOWN X                         .ONE  130
C .                                                                        .ONE  140
C ..........................................................................ONE  150
C                                                                           ONE  160
```

```
      DIMENSION XNOW(NDEC,1), DELTAX(NDEC)                                  ONE  170
      DIMENSION GNOW(30,1), GNEXT(30,1), P(30,1), Q(30,1), PQ(1,1)          ONE  180
      DIMENSION QSQ(1,1), PP(30,30), SQQS(30,30), S(30,30)                  ONE  190
      DIMENSION XTEMP(30,1), PT(1,30), QS(1,30), SQ(1,30), QT(1,30)         ONE  200
      DIMENSION SQQ(30,30)                                                  ONE  210
      DIMENSION D(30,1)                                                     ONE  220
      IT = 0                                                                ONE  230
      CALL COST1 (0,N,FNOW,XNOW,ANII,PNALTY,CLOSE,THRESH,NDEC)              ONE  240
      IF (PNALTY.GE.PERCRIT) GO TO 10                                       ONE  250
      CALL MONIT (IT,XNOW,N,FNOW,ANII,PNALTY,CLOSE,THRESH,NDEC)             ONE  260
      IF (MAXIT.EQ.0) STOP                                                  ONE  270
      GO TO 20                                                              ONE  280
   10 WRITE (6,9020) PNALTY,ANII                                            ONE  290
      CALL CORNER (XNOW,N,NDEC)                                             ONE  300
      IF (MAXIT.EQ.0) STOP                                                  ONE  310
C                                                                           ONE  320
C ..........................................................................ONE  330
C .                                                                        . ONE  340
C .   STEP 1:  SET S = IDENTITY MATRIX AND CALCULATE GRADIENT G            . ONE  350
C .                                                                        . ONE  360
C ..........................................................................ONE  370
C                                                                           ONE  380
   20 DO 30 I = 1,N                                                         ONE  390
        DO 30 J = 1,N                                                       ONE  400
        S(I,J) = 0.                                                         ONE  410
        IF (I.EQ.J) S(I,J) = 1.                                             ONE  420
   30   CONTINUE                                                            ONE  430
      CALL FGRAD (N,FNOW,XNOW,GNOW,DELTAX,NDEC)                             ONE  440
C                                                                           ONE  450
C ..........................................................................ONE  460
C .                                                                        . ONE  470
C .   STEP 2:  SET D = -SG                                                 . ONE  480
C .                                                                        . ONE  490
C ..........................................................................ONE  500
C                                                                           ONE  510
   40 CALL MPLY (N,N,1,S,GNOW,D,NDEC,NDEC,1)                                ONE  520
      DO 50 I = 1,N                                                         ONE  530
        D(I,1) = -D(I,1)                                                    ONE  540
   50   CONTINUE                                                            ONE  550
C                                                                           ONE  560
C ..........................................................................ONE  570
C .                                                                        . ONE  580
C .   STEP 3:  LINE SEARCH ALONG D TO FIND AFA THAT SATISFIES PQ>0         . ONE  590
C .                                                                        . ONE  600
C ..........................................................................ONE  610
C                                                                           ONE  620
      K = 0                                                                 ONE  630
   60 K = K+1                                                               ONE  640
      CALL LINESCH (K,N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,CLOSE,ONE  650
     $THRESH,NDEC)                                                          ONE  660
      CALL FGRAD (N,FSMALL,XTEMP,GNEXT,DELTAX,NDEC)                         ONE  670
      DO 70 I = 1,N                                                         ONE  680
        Q(I,1) = GNEXT(I,1)-GNOW(I,1)                                       ONE  690
        P(I,1) = AFA2*D(I,1)                                                ONE  700
   70   CONTINUE                                                            ONE  710
      CALL TRNSPOS (NDEC,1,P,PT)                                            ONE  720
      CALL MPLY (1,N,1,PT,Q,PQ,1,NDEC,1)                                    ONE  730
      IF (PQ(1,1).GT.0.) GO TO 90                                           ONE  740
      IF (K.GT.3) RETURN                                                    ONE  750
      FNOW = FSMALL                                                         ONE  760
      DO 80 I = 1,N                                                         ONE  770
        XNOW(I,1) = XTEMP(I,1)                                              ONE  780
   80   CONTINUE                                                            ONE  790
      GO TO 60                                                              ONE  800
```

```
   90 DO 100 I = 1,N                                                    ONE  P10
      . XNOW(I,1) = XTEMP(I,1)                                          ONE  P20
  100   CONTINUE                                                       ONE  P30
      IT = IT+1                                                         ONE  P40
      PRCNT = ABS((FSMALL-FNOW)/FNOW)                                   ONE  P50
      IF (PRCNT.GE.STOPCHG) GO TO 120                                   ONE  P60
      WRITE (6,9030) PRCNT,STOPCHG                                      ONE  P70
      IF (PNALTY.GE.PENCRIT) GO TO 110                                  ONE  P80
      CALL MONIT (IT,XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH,NDEC)       ONE  P90
C                                                                       ONE  900
      RETURN                                                            ONE  910
  110 WRITE (6,9010) PNALTY                                             ONE  920
      CALL CORNER (XNOW,N,NDEC)                                         ONE  930
      RETURN                                                            ONE  940
  120 IF (IT.LT.MAXIT) GO TO 140                                        ONE  950
      WRITE (6,9040)                                                    ONE  960
      IF (PNALTY.GE.PENCRIT) GO TO 130                                  ONE  970
      CALL MONIT (IT,XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH,NDEC)       ONE  980
C                                                                       ONE  990
      RETURN                                                            ONE 1000
  130 WRITE (6,9010) PNALTY                                             ONE 1010
      CALL CORNER (XNOW,N,NDEC)                                         ONE 1020
      RETURN                                                            ONE 1030
  140 IF (PNALTY.GE.PENCRIT) GO TO 150                                  ONE 1040
      CALL MONIT (IT,XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH,NDEC)       ONE 1050
      GO TO 160                                                         ONE 1060
  150 WRITE (6,9020) PNALTY,ANII                                        ONE 1070
      CALL CORNER (XNOW,N,NDEC)                                         ONE 1080
  160 CONTINUE                                                          ONE 1090
C                                                                       ONE 1100
C ..................................................................... ONE 1110
C .                                                                   . ONE 1120
C .   STEP 4:  IF "IT" IS INTEGER MULTIPLE OF N GO TO STEP 1;         . ONE 1130
C .         IF NOT, UPDATE S                                          . ONE 1140
C .     IS "IT" INTEGER MULTIPLE OF N?                                . ONE 1150
C .                                                                   . ONE 1160
C ..................................................................... ONE 1170
C                                                                       ONE 1180
      IF ((FLOAT(IT)/FLOAT(N)).NE.FLOAT(IT/N)) GO TO 170                ONE 1190
      FNOW = FSMALL                                                     ONE 1200
      GO TO 20                                                          ONE 1210
C                                                                       ONE 1220
C ..................................................................... ONE 1230
C .                                                                   . ONE 1240
C .     UPDATE MATRIX S; GO TO STEP 2                                 . ONE 1250
C .                                                                   . ONE 1260
C ..................................................................... ONE 1270
C                                                                       ONE 1280
  170 CALL MPLY (N,N,1,S,G,SG,NDEC,NDEC,1)                              ONE 1290
      CALL TRNSPOS (NDEC,1,G,GT)                                        ONE 1300
      CALL MPLY (N,1,N,SG,GT,SGG,NDEC,1,NDEC)                           ONE 1310
      CALL MPLY (N,N,N,SGG,S,SGGS,NDEC,NDEC,NDEC)                       ONE 1320
      CALL MPLY (1,N,N,GT,S,GS,1,NDEC,NDEC)                             ONE 1330
      CALL MPLY (1,N,1,GS,G,GSG,1,NDEC,1)                               ONE 1340
      CALL MPLY (N,1,N,P,PT,PP,NDEC,1,NDEC)                             ONE 1350
      DO 180 I = 1,N                                                    ONE 1360
        DO 180 J = 1,N                                                  ONE 1370
        S(I,J) = (S(I,J)-SGGS(I,J)/GSG(1,1))*(PG(1,1)/GSG(1,1))+PP(I,J)/ONE 1380
     $  PG(1,1)                                                         ONE 1390
  180   CONTINUE                                                       ONE 1400
      FNOW = FSMALL                                                     ONE 1410
      DO 190 I = 1,N                                                    ONE 1420
        GNOW(I,1) = GNEXT(I,1)                                          ONE 1430
  190   CONTINUE                                                       ONE 1440
```

```
      GO TO 40                                                          ONE 1450
C     .                                                                 ONE 1460
 9010 FORMAT (5X,40HIN RESULT: DYNAMIC CONSTRAINTS VIOLATION,//,10X,9HPENONE 1470
     $ALTY =,1PE16.9)                                                   ONE 1480
 9020 FORMAT (5X,39HIN MONIT: DYNAMIC CONSTRAINTS VIOLATION,//,10X,9HPENONE 1490
     $LTY =,1PE16.9,//,10X,"ANII = ",E16.9," (IF = 0.,  THEN ANII HAS",ONONE 1500
     $T BEEN COMPUTED IN THIS ITERATION, DUE TO LARGE PNALTY")          ONE 1510
 9030 FORMAT (2X,"RELATIVE CHANGE IN SUCCESSIVE COSTS ",1PE10.3," LESS "ONE 1520
     $," THAN STOP CRITERION ",1PE10.3)                                 ONE 1530
 9040 FORMAT (2X,29HMAXIMUM ITERATION SET REACHED)                      ONE 1540
      END                                                               ONE 1550


      SUBROUTINE MPLY (L,M,N,A,B,C,LDEC,MDEC,NDEC)                      MPL   10
C                                                                       MPL   20
C ***********************************************************************MPL   30
C .                                                                   . MPL   40
C .   CALCULATE MATRIX MULTIPLICATION C = AB                          . MPL   50
C .                                                                   . MPL   60
C ***********************************************************************MPL   70
C                                                                       MPL   80
      DIMENSION A(LDEC,MDEC), B(MDEC,NDEC), C(LDEC,NDEC)               MPL   90
      DO 10 I = 1,L                                                     MPL  100
        DO 10 J = 1,N                                                   MPL  110
        C(I,J) = 0.                                                     MPL  120
        DO 10 K = 1,M                                                   MPL  130
        C(I,J) = C(I,J)+A(I,K)*B(K,J)                                   MPL  140
   10   CONTINUE                                                        MPL  150
      RETURN                                                            MPL  160
      END                                                               MPL  170


      SUBROUTINE TRNSPOS (M,N,A,B)                                      TRN   10
C                                                                       TRN   20
C ***********************************************************************TRN   30
C .                                                                   . TRN   40
C .   TRANSPOSE OF MATRIX A IS RETURNED IN MATRIX B                   . TRN   50
C .                                                                   . TRN   60
C ***********************************************************************TRN   70
C                                                                       TRN   80
      DIMENSION A(M,N), B(N,M)                                          TRN   90
      DO 10 I = 1,M                                                     TRN  100
        DO 10 J = 1,N                                                   TRN  110
        B(J,I) = A(I,J)                                                 TRN  120
   10   CONTINUE                                                        TRN  130
      RETURN                                                            TRN  140
      END                                                               TRN  150


      SUBROUTINE FGRAD (N,F,X,G,DELTAX,NDEC)                            FGR   10
C                                                                       FGR   20
C ***********************************************************************FGR   30
C .                                                                   . FGR   40
C .   CALCULATE GRADIENT OF COST F WITH RESPECT TO UNKNOWN X          . FGR   50
C .                                                                   . FGR   60
C ***********************************************************************FGR   70
C                                                                       FGR   80
      DIMENSION X(NDEC,1), G(NDEC,1), DELTAX(NDEC,1)                    FGR   90
```

```
      DO 10 I = 1,N                                                    FGR  100
        G(I,1) = 0.                                                    FGR  110
        IF (MOD(I,3).EQ.0) GO TO 10                                    FGR  120
        X(I,1) = X(I,1)+DELTAX(I)                                      FGR  130
        CALL COST1 (1,N,FF,X,ANII,PNALTY,CLOSE,THRESH,NDEC)            FGR  140
        G(I,1) = (FF-F)/DELTAX(I)                                      FGR  150
        X(I,1) = X(I,1)-DELTAX(I)                                      FGR  160
   10   CONTINUE                                                       FGR  170
      RETURN                                                           FGR  180
      END                                                              FGR  190


      SUBROUTINE GAFA (N,F,DAFA,G,X,D,NDEC)                            GAF   10
C                                                                      GAF   20
C ..................................................................   GAF   30
C .                                                                .   GAF   40
C .   CALCULATE GRADIENT OF F WITH RESPECT TO AFA                  .   GAF   50
C .                                                                .   GAF   60
C ..................................................................   GAF   70
C                                                                      GAF   80
      DIMENSION X(NDEC,1), D(NDEC,1), XTEMP(30,1)                      GAF   90
      DO 10 I = 1,N                                                    GAF  100
        XTEMP(I,1) = X(I,1)+DAFA*D(I,1)                                GAF  110
   10   CONTINUE                                                       GAF  120
      CALL COST1 (1,N,FF,XTEMP,ANII,PNALTY,CLOSE,THRESH)               GAF  130
      G = (FF-F)/DAFA                                                  GAF  140
      RETURN                                                           GAF  150
      END                                                              GAF  160


      SUBROUTINE ERROR (K,MESSAGE)                                     ERO   10
      DIMENSION MESSAGE(12), ITEXT(12)                                 ERO   20
      DATA ITEXT/12*" "/                                               ERO   30
      DO 10 I = 1,K                                                    ERO   40
   10   ITEXT(I) = MESSAGE(I)                                          ERO   50
      WRITE (6,9010) (ITEXT(I),I=1,K)                                  ERO   60
      STOP                                                             ERO   70
C                                                                      ERO   80
 9010 FORMAT (///,1X,"ERROR: ",12A10,///)                              ERO   90
      END                                                              ERO  100


      SUBROUTINE LINESCH (K,N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,LIN   10
     1CLOSE,THRESH,NDEC)                                               LIN   20
      DIMENSION D(NDEC,1), XTEMP(NDEC,1), XNOW(NDEC,1)                 LIN   30
C                                                                      LIN   40
C ..................................................................   LIN   50
C .                                                                .   LIN   60
C .   CUBIC FIT BY INITIALLY LETTING AFA = 0 AND DAFA = 5.         .   LIN   70
C .                                                                .   LIN   80
C ..................................................................   LIN   90
C                                                                      LIN  100
      DNORM = 0.                                                       LIN  110
      DO 10 I = 1,N                                                    LIN  120
        DNORM = DNORM+D(I,1)**2                                        LIN  130
   10   CONTINUE                                                       LIN  140
      DNORM = SQRT(DNORM)                                              LIN  150
      IF (DNORM.EQ.0.) CALL CHECK (1,N,FNOW,XNOW,D,AFA0,FAFA0,GAFA0,AFA1LIN  160
```

```
     $,FAFA1,GAFA1,U1,UU1,U2,NDEC)                                          LIN  170
C                                                                           LIN  180
C ..........................................................................LIN  190
C .                                                                        .LIN  200
C .   NORMALIZE D                                                         .LIN  210
C .                                                                        .LIN  220
C ..........................................................................LIN  230
C                                                                           LIN  240
      DO 20 I = 1,N                                                         LIN  250
        D(I,1) = D(I,1)/DNORM                                               LIN  260
   20   CONTINUE                                                           LIN  270
      DAFA = 5.                                                             LIN  280
      AFA0 = 0.                                                             LIN  290
      FAFA0 = FNOW                                                          LIN  300
      KK = 1                                                                LIN  310
      CALL GAFA (N,FAFA0,DAFA,GAFA0,XNOW,D,NDEC)                            LIN  320
      IF (GAFA0.EQ.0) CALL CHECK (1,N,FNOW,XNOW,D,AFA0,FAFA0,GAFA0,AFA1,LIN  330
     $FAFA1,GAFA1,U1,UU1,U2,NDEC)                                           LIN  340
      IF (-FAFA0/GAFA0.LE.0.) CALL ERROR (6," -FAFA0/GAFA0 .LE. 0., TRAJ LIN  350
     $. SET PROBABLY NEAR MINIMUM PT.")                                     LIN  360
      AFA1 = AMIN1(1000.,-FAFA0/GAFA0)                                      LIN  370
   30 DO 40 I = 1,N                                                         LIN  380
        XTEMP(I,1) = XNOW(I,1)+AFA1*D(I,1)                                  LIN  390
   40   CONTINUE                                                           LIN  400
      CALL COST1 (1,N,FAFA1,XTEMP,ANII,PNALTY,CLOSE,THRESH,NDEC)            LIN  410
      IF (KK.EQ.4) CALL ERROR (7,"PROBLEM IN LINESCH.  CAN'T FIND LOWER LIN  420
     $COST AFTER 3 CUBIC FITS")                                             LIN  430
      CALL GAFA (N,FAFA1,DAFA,GAFA1,XTEMP,D,NDEC)                           LIN  440
      IF (ABS((AFA0-AFA1)/AFA1).LT..001) CALL CHECK (2,N,FNOW,XNOW,D,AFALIN  450
     $0,FAFA0,GAFA0,AFA1,FAFA1,GAFA1,U1,UU1,U2,NDEC)                        LIN  460
      U1 = GAFA0+GAFA1-3.*(FAFA0-FAFA1)/(AFA0-AFA1)                         LIN  470
      UU1 = U1**2-GAFA0*GAFA1                                               LIN  480
      IF (UU1.LE.0.) U2 = 0.                                                LIN  490
      IF (UU1.GT.0.) U2 = SQRT(UU1)                                         LIN  500
      A0 = AMIN1(AFA0,AFA1)                                                 LIN  510
      A1 = AMAX1(AFA0,AFA1)                                                 LIN  520
      G0 = GAFA0                                                            LIN  530
      F0 = FAFA0                                                            LIN  540
      G1 = GAFA1                                                            LIN  550
      F1 = FAFA1                                                            LIN  560
      IF (AFA0.LT.AFA1) GO TO 50                                            LIN  570
      G0 = GAFA1                                                            LIN  580
      F0 = FAFA1                                                            LIN  590
      G1 = GAFA0                                                            LIN  600
      F1 = FAFA0                                                            LIN  610
   50 IF (G1-G0+2.*U2.EQ.0.) CALL CHECK (3,N,FNOW,XNOW,D,A0,F0,G0,A1,F1,LIN  620
     $G1,U1,UU1,U2,NDEC)                                                    LIN  630
      AFA2 = A1-(A1-A0)*(G1+U2-U1)/(G1-G0+2.*U2)                            LIN  640
      DO 60 I = 1,N                                                         LIN  650
        XTEMP(I,1) = XNOW(I,1)+AFA2*D(I,1)                                  LIN  660
   60   CONTINUE                                                           LIN  670
      CALL COST1 (0,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH,NDEC)           LIN  680
      WRITE (6,9010) A0,A1,AFA2,DNORM,F0,F1,FSMALL,G0,G1                    LIN  690
      WRITE (6,9020) (D(I,1),I=1,N)                                         LIN  700
      IF (FSMALL.GE.FNOW) GO TO 70                                          LIN  710
      RETURN                                                                LIN  720
   70 AFA1 = AFA1/2.                                                        LIN  730
      KK = KK+1                                                             LIN  740
      GO TO 30                                                              LIN  750
C                                                                           LIN  760
 9010 FORMAT (15X,"A0 = ",1PE13.6,5X,"A1 = ",E13.6,5X,"AFA2 = ",E13.6,5XLIN  770
     $,"DNORM = ",E13.6,/,15X,"F0 = ",E13.6,5X,"F1 = ",E13.6,5X,"FSMALL LIN  780
     $= ",E13.6,/,15X,"G0 = ",E13.6,5X,"G1 = ",E13.6,/)                     LIN  790
 9020 FORMAT (5X,"D = ",10(1X,1PE9.2,2X))                                   LIN  800
```

```
      END                                                                LIN  810


      SUBROUTINE CHECK (ICHECK,N,FNOW,XNOW,D,A0,F0,G0,A1,F1,G1,U1,UU1,U2CHE   10
     $,NDEC)                                                            CHE   20
      DIMENSION XNOW(NDEC,1), D(NDEC,1)                                 CHE   30
      WRITE (6,10) FNOW,N,(I,XNOW(I,1),D(I,1),I=1,N)                    CHE   40
      IF (ICHECK.EQ.1) STOP                                             CHE   50
      WRITE (6,20) A0,F0,G0,A1,F1,G1                                    CHE   60
      IF (ICHECK.EQ.2) STOP                                             CHE   70
      WRITE (6,30) U1,UU1,U2                                            CHE   80
      STOP                                                              CHE   90
C                                                                       CHE  100
   10 FORMAT (20X,44HTHIS IS SUBROUTINE CHECK WHICH GIVES ALL THE,35H INCHE  110
     $FORMATION IN SUBROUTINE LINESCH.,///,30X,7HFNOW = ,1PE16.9,///,37X,4CHE  120
     $HXNOW,17X,1HD,///,(25X,I2,3X,1PE16.9,4X,1PE16.9,/))               CHE  130
   20 FORMAT (///,20X,"A0 = ",1PE16.9,5X,"F0 = ",E16.9,5X,"G0 = ",E16.9,/CHE  140
     $20X,"A1 = ",E16.9,5X,"F1 = ",E16.9,5X,"G1 = ",E16.9)              CHE  150
   30 FORMAT (///,20X,5HU1 = ,1E16.9,5X,6HUU1 = ,1PE16.9,5X,5HU2 = ,1PE16CHE  160
     $.9)                                                               CHE  170
      END                                                               CHE  180


      SUBROUTINE CORNER (X,N,NDEC)                                      CCR   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITCCR   20
     $(3,200,3),XC(3),XF(3),YC(3),YF(3),ZC(3),ZF(3),NPOSITS(3),RHO(576),CCR   30
     $ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,FATOFC,FATOFW         CCR   40
      DIMENSION X(NDEC,1)                                               CCR   50
      NVAR = 2                                                          CCR   60
      IF (IPROF.EQ.1) NVAR = 3                                          CCR   70
      DO 10 I = 1,NTRAJ                                                 CCR   80
        NSEG1 = NSEG-1                                                  CCR   90
        DO 10 J = 1,NSEG1                                               CCR  100
        DO 10 K = 1,NVAR                                                CCR  110
        L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                               CCR  120
        IF (K.EQ.1) XM(I,J+1) = X(L,1)                                  CCR  130
        IF (K.EQ.2) YM(I,J+1) = X(L,1)                                  CCR  140
        IF (K.EQ.3) ZM(I,J+1) = X(L,1)                                  CCR  150
   10   CONTINUE                                                        CCR  160
      DO 30 I = 1,NTRAJ                                                 CCR  170
        WRITE (6,9010) I,XM(I,1),YM(I,1),ZM(I,1)                        CCR  180
        DO 20 J = 2,NSEG                                                CCR  190
          WRITE (6,9020) XM(I,J),YM(I,J),ZM(I,J)                        CCR  200
   20     CONTINUE                                                      CCR  210
        WRITE (6,9020) XM(I,NSEG+1),YM(I,NSEG+1),ZM(I,NSEG+1)           CCR  220
   30   CONTINUE                                                        CCR  230
      RETURN                                                            CCR  240
C                                                                       CCR  250
 9010 FORMAT (12X,"FLIGHT PATH NO: ",I1,/,14X,"CORNER PTS.",/,14X,"(",F7CCR  260
     $.0,",",F7.0,",",F7.0,")")                                         CCR  270
 9020 FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")")                       CCR  280
      END                                                               CCR  290


      SUBROUTINE DYNARID (PNALTY)                                       DYN   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITDYN   20
     $(3,200,3),XC(3),XF(3),YC(3),YF(3),ZC(3),ZF(3),NPOSITS(3),RHO(576),DYN   30
     $ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,FATOFC,FATOFW         DYN   40
```

```
      COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,ZWEIT1,GMAMIN,GMAMAX,NSAMP  DYN   50
     . COMMON /CORNER/ ANGLE(3,3),POSIT(3,200,3),XCENTR(3,3),YCENTR(3,3),DYN   60
     1ZCENTR(3,3),XT1(3,5),YT1(3,5),ZT1(3,5),XT2(3,5),YT2(3,5),ZT2(3,5),DYN   70
     2NPOSIT(3),DIS(3)                                                  DYN   80
      REAL LENGTH                                                       DYN   90
      LENGTH(A,B,C,D,E,F) = SQRT((A-B)**2+(C-D)**2+(E-F)**2)            DYN  100
      PI = ATAN(1.)*4.                                                  DYN  110
C                                                                       DYN  120
C .....................................................................  DYN  130
C .                                                                   .  DYN  140
C .   FIPST CALCULATE PARAMETERS AT CORNER POINTS                     .  DYN  150
C .       "                                                           .  DYN  160
C .....................................................................  DYN  170
C                                                                       DYN  180
      DO 60 I = 1,NTRAJ                                                 DYN  190
        DO 10 J = 2,NSEG                                                DYN  200
          YT2(I,J) = 0.                                                 DYN  210
          YT1(I,J) = YT2(I,J)                                           DYN  220
          XT2(I,J) = YT1(I,J)                                           DYN  230
          XT1(I,J) = XT2(I,J)                                           DYN  240
          ZT1(I,J) = 0.                                                 DYN  250
          ZT2(I,J) = 0.                                                 DYN  260
          ANGLE(I,J-1) = XT1(I,J)                                       DYN  270
          YCENTR(I,J-1) = ANGLE(I,J-1)                                  DYN  280
          XCENTR(I,J-1) = YCENTR(I,J-1)                                 DYN  290
   10     CONTINUE                                                      DYN  300
        DIS(I) = 0.                                                     DYN  310
        XT1(I,1) = XM(I,1)                                              DYN  320
        XT2(I,1) = XM(I,1)                                              DYN  330
        YT1(I,1) = YM(I,1)                                              DYN  340
        YT2(I,1) = YM(I,1)                                              DYN  350
        ZT1(I,1) = ZM(I,1)                                              DYN  360
        ZT2(I,1) = ZM(I,1)                                              DYN  370
        ISEG2 = ISEG1+1                                                 DYN  380
        DO 40 J = ISEG2,NSEG                                            DYN  390
          A = LENGTH(XM(I,J-1),XM(I,J),YM(I,J-1),YM(I,J),ZM(I,J-1),ZM(I,DYN  400
     $    J))                                                           DYN  410
          B = LENGTH(XM(I,J),XM(I,J+1),YM(I,J),YM(I,J+1),ZM(I,J),ZM(I,J+DYN  420
     $    1))                                                           DYN  430
          C = LENGTH(XM(I,J-1),XM(I,J+1),YM(I,J-1),YM(I,J+1),ZM(I,J-1),ZDYN  440
     $    M(I,J+1))                                                     DYN  450
C                                                                       DYN  460
C .....................................................................  DYN  470
C .                                                                   .  DYN  480
C .   DO THE TWO SEGMENTS FORM A STRAIGHT LINE?                       .  DYN  490
C .                                                                   .  DYN  500
C .....................................................................  DYN  510
C                                                                       DYN  520
          IF (ABS(A+B-C).GT.1.) GO TO 20                                DYN  530
          SEG = LENGTH(XT2(I,J-1),XM(I,J),YT2(I,J-1),YM(I,J),ZT2(I,J-1),DYN  540
     $    ZM(I,J))                                                      DYN  550
          XT1(I,J) = XM(I,J)                                            DYN  560
          XT2(I,J) = XM(I,J)                                            DYN  570
          YT1(I,J) = YM(I,J)                                            DYN  580
          YT2(I,J) = YM(I,J)                                            DYN  590
          ZT1(I,J) = ZM(I,J)                                            DYN  600
          ZT2(I,J) = ZM(I,J)                                            DYN  610
          ZCENTR(I,J-1) = 0.                                            DYN  620
          YCENTR(I,J-1) = 0.                                            DYN  630
          XCENTR(I,J-1) = YCENTR(I,J-1)                                 DYN  640
          ALONG1 = 0.                                                   DYN  650
          ANGLE(I,J-1) = C.                                             DYN  660
          GO TO 30                                                      DYN  670
   20     AXY = LENGTH(XM(I,J-1),XM(I,J),YM(I,J-1),YM(I,J),0.,0.)       DYN  680
```

```
      BXY = LENGTH(XM(I,J),XM(I,J+1),YM(I,J),YM(I,J+1),0.,0.)          DYN  690
      CXY = LENGTH(XM(I,J-1),XM(I,J+1),YM(I,J-1),YM(I,J+1),0.,0.)      DYN  700
      BETA = AACOS(AXY,BXY,CXY)                                        DYN  710
      ALONG2 = RADIUS/TAN(BETA/2.)                                     DYN  720
      PNALTY = PNALTY+PWEIT1*((AMAX1(0.,(ALONG2-AXY)))**2+AMAX1(0.,(DYN  730
     $ ALONG2-BXY)))**2                                                DYN  740
      IF (J.NE.ISEG2) PNALTY = PNALTY+PWEIT1*(AMAX1(0.,(ALONG1+ALONGDYN  750
     $ 2-AXY)))**2                                                     DYN  760
C                                                                      DYN  770
C .....................................................................DYN  780
C .                                                                   .DYN  790
C .  CALCULATE TANGENTIAL POINTS, CENTER OF RADIUS R AT J-TH CORNER   .DYN  800
C .                                                                   .DYN  810
C .....................................................................DYN  820
C                                                                      DYN  830
      XT1(I,J) = XM(I,J)+ALONG2*(XM(I,J-1)-XM(I,J))/AXY                DYN  840
      YT1(I,J) = YM(I,J)+ALONG2*(YM(I,J-1)-YM(I,J))/AXY                DYN  850
      XT2(I,J) = XM(I,J)+ALONG2*(XM(I,J+1)-XM(I,J))/BXY                DYN  860
      YT2(I,J) = YM(I,J)+ALONG2*(YM(I,J+1)-YM(I,J))/BXY                DYN  870
      ZT1(I,J) = ZM(I,J)+ALONG2*(ZM(I,J-1)-ZM(I,J))/AXY                DYN  880
      ZT2(I,J) = ZM(I,J)+ALONG2*(ZM(I,J+1)-ZM(I,J))/BXY                DYN  890
      SEG = LENGTH(XT2(I,J-1),XT1(I,J),YT2(I,J-1),YT1(I,J),ZT2(I,J-1)DYN  900
     $ ),ZT1(I,J))                                                     DYN  910
      DISM2 = ALONG2*COS(BETA/2.)                                      DYN  920
      DISC2 = RADIUS/SIN(BETA/2.)                                      DYN  930
      X = (XT1(I,J)+XT2(I,J))/2.                                       DYN  940
      Y = (YT1(I,J)+YT2(I,J))/2.                                       DYN  950
      XCENTR(I,J-1) = XM(I,J)+DISC2*(X-XM(I,J))/DISM2                  DYN  960
      YCENTR(I,J-1) = YM(I,J)+DISC2*(Y-YM(I,J))/DISM2                  DYN  970
      ALONG1 = ALONG2                                                  DYN  980
      D = LENGTH(XT1(I,J),XT2(I,J),YT1(I,J),YT2(I,J),0.,0.)            DYN  990
      ANGLE(I,J-1) = AACOS(RADIUS,RADIUS,D)                            DYN 1000
      SEG = SEG+ANGLE(I,J-1)*RADIUS                                    DYN 1010
   30 DIS(I) = DIS(I)+SEG                                              DYN 1020
   40 CONTINUE                                                         DYN 1030
      DO 50 J = ISEG2,NSEG                                             DYN 1040
C                                                                      DYN 1050
C .....................................................................DYN 1060
C .                                                                   .DYN 1070
C .  CALCULATE VERTICAL SLOPE OF EACH SEGMENT AND                     .DYN 1080
C .  ADD PENALTY FOR EACH SEGMENT WITH VERTICAL SLOPE GREATER         .DYN 1090
C .  THAN GMAMAX OR LESS THAN GMAMIN DEGREES                          .DYN 1100
C .                                                                   .DYN 1110
C .....................................................................DYN 1120
C                                                                      DYN 1130
      XYL = LENGTH(XM(I,J+1),XM(I,J),YM(I,J+1),YM(I,J),0.,0.)          DYN 1140
      VSLOPE = (ZM(I,J+1)-ZM(I,J))/XYL                                 DYN 1150
      PNALTY = PNALTY+ZWEIT1*(AMAX1(0.,VSLOPE-TAND(GMAMAX)))**2+ZWEIDYN 1160
     $ T1*(AMAX1(0.,TAND(GMAMIN)-VSLOPE))**2                           DYN 1170
   50 CONTINUE                                                         DYN 1180
      AXY = LENGTH(XM(I,NSEG),XM(I,NSEG+1),YM(I,NSEG),YM(I,NSEG+1),0.,DYN 1190
     $ 0.)                                                             DYN 1200
      BXY = LENGTH(XM(I,NSEG+1),XM(I,NSEG+2),YM(I,NSEG+1),YM(I,NSEG+2)DYN 1210
     $ ,0.,0.)                                                         DYN 1220
      CXY = LENGTH(XM(I,NSEG),XM(I,NSEG+2),YM(I,NSEG),YM(I,NSEG+2),0.,DYN 1230
     $ 0.)                                                             DYN 1240
      ANG = AACOS(AXY,BXY,CXY)                                         DYN 1250
      PNALTY = PNALTY+PWEIT1*(AMAX1(0.,(400.+ALONG1-AXY)))**2+PWEIT2*(DYN 1260
     $ AMAX1(0.,(2.3605-ANG)))**2                                      DYN 1270
      DIS(I) = DIS(I)+LENGTH(XT2(I,NSEG),XM(I,NSEG+1),YT2(I,NSEG),YM(IDYN 1280
     $ ,NSEG+1),ZT2(I,NSEG),ZM(I,NSEG+1))                              DYN 1290
      FATOFC = FATOFC+FATOFW*DIS(I)                                    DYN 1300
   60 CONTINUE                                                         DYN 1310
      RETURN                                                           DYN 1320
```

```
      END                                                                DYN 1330


      SUBROUTINE NII (ANII)                                              NII   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITNII   20
     $(3,200,3),XO(3),XF(3),YO(3),YF(3),ZO(3),ZF(3),NPOSITS(3),RHO(576),NII   30
     $ISEG1,IPPOF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,FATOFC,FATOFW          NII   40
      PEOPLE = 0.                                                        NII   50
      ALWP = 0.                                                          NII   60
      DO 10 K = 1,NMAP                                                   NII   70
        IF (ARRAY(K,3).LT.1.) GO TO 10                                   NII   80
        ALON = ARRAY(K,4)                                                NII   90
        W = 3.36E-6*10.**(0.103*ALON)                                    NII  100
        W = W/(0.2*10.**(0.03*ALON)+1.43E-4*10.**(0.08*ALON))            NII  110
        PEOPLE = PEOPLE+ARRAY(K,3)                                       NII  120
        ARRAY(K,5) = ARRAY(K,3)*W                                        NII  130
        ALWP = ALWP+ARRAY(K,5)                                           NII  140
   10   CONTINUE                                                         NII  150
      ANII = ALWP/PEOPLE*AWEIT                                           NII  160
      RETURN                                                             NII  170
      END                                                                NII  180


      SUBROUTINE NCISEL (THRESH)                                         NCI   10
      COMMON NTRAJ,NMAP,NSEG,XM(3,6),YM(3,6),ZM(3,6),ARRAY(576,9),SPOSITNCI   20
     $(3,200,3),XO(3),XF(3),YO(3),YF(3),ZO(3),ZF(3),NPOSITS(3),RHO(576),NCI   30
     $ISEG1,IPPOF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,FATOFC,FATOFW          NCI   40
      COMMON /CORNER/ ANGLE(3,3),POSIT(3,200,3),XCENTR(3,3),YCENTR(3,3),NCI   50
     $ZCENTR(3,3),XT1(3,5),YT1(3,5),ZT1(3,5),XT2(3,5),YT2(3,5),ZT2(3,5),NCI   60
     $NPOSIT(3),DIS(3)                                                   NCI   70
      COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,ZWEIT1,GMAMIN,GMAMAX,NSAMP   NCI   80
      COMMON /THRESH/ ALMAX,MAXFLIT,TWEIT1,TWEIT2,NPLANE                 NCI   90
      COMMON /SAMPLE/ SAMPT,PERIOD,SAMLTH,NAC(3,32),CNCIS(16,2)          NCI  100
      DIMENSION ARCLTH(4), SEGLTH(5), AL(16)                             NCI  110
      REAL LENGTH                                                        NCI  120
      COORD(A,B,C,D) = (A/B)*(C-D)+D                                     NCI  130
      LENGTH(A,B,C,D,E,F) = SQRT((A-B)**2+(C-D)**2+(E-F)**2)             NCI  140
      PI = ATAN(1.)*4.                                                   NCI  150
      DO 100 I = 1,NTRAJ                                                 NCI  160
        XT2(I,NSEG+1)=XM(I,NSEG+1)                                       NCI  170
        XT1(I,NSEG+1)=XT2(I,NSEG+1)                                      NCI  180
        YT2(I,NSEG+1)=YM(I,NSEG+1)                                       NCI  190
        YT1(I,NSEG+1)=YT2(I,NSEG+1)                                      NCI  200
        ZT2(I,NSEG+1)=ZM(I,NSEG+1)                                       NCI  210
        ZT1(I,NSEG+1)=ZT2(I,NSEG+1)                                      NCI  220
        DLZPM = (ZO(I)-ZF(I))/DIS(I)                                     NCI  230
        POSIT(I,1,1) = XM(I,ISEG1)                                       NCI  240
        POSIT(I,1,2) = YM(I,ISEG1)                                       NCI  250
        POSIT(I,1,3) = ZM(I,ISEG1)                                       NCI  260
        NSEG1 = NSEG-1                                                   NCI  270
        DO 10 J = ISEG1,NSEG1                                            NCI  280
          SEGLTH(J) = LENGTH(XT2(I,J),XT1(I,J+1),YT2(I,J),YT1(I,J+1),ZT2NCI  290
     $    (I,J),ZT1(I,J+1))                                              NCI  300
          ARCLTH(J) = RADIUS*ANGLE(I,J) *                                NCI  310
   10     CONTINUE                                                       NCI  320
        SEGLTH(NSEG) = LENGTH(XT2(I,NSEG),XM(I,NSEG+1),YT2(I,NSEG),YM(I,NCI  330
     $  NSEG+1),ZT2(I,NSEG),ZM(I,NSEG+1))                               NCI  340
        II = 1                                                           NCI  350
        DO 70 J = ISEG1,NSEG1                                            NCI  360
   20     IF (SEGLTH(J).LT.SAMLTH) GO TO 30                             NCI  370
```

```
      II = II+1                                                          NCI  380
     .POSIT(I,II,1) = COORD(SAMLTH,SEGLTH(J),XT1(I,J+1),POSIT(I,II-1)NCI  390
     $,1))                                                               NCI  400
      POSIT(I,II,2) = COORD(SAMLTH,SEGLTH(J),YT1(I,J+1),POSIT(I,II-1)NCI  410
     $,2))                                                               NCI  420
      POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI  430
      IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(SAMLTH,SEGLTH(J),ZT1(I,JNCI  440
     $+1),POSIT(I,II-1,3))                                               NCI  450
      SEGLTH(J) = SEGLTH(J)-SAMLTH                                       NCI  460
      GO TO 20                                                           NCI  470
30    IF (SEGLTH(J)+ARCLTH(J).GE.SAMLTH) GO TO 40                        NCI  480
      PIECE = SAMLTH-ARCLTH(J)-SEGLTH(J)                                 NCI  490
      II = II+1                                                          NCI  500
      POSIT(I,II,1) = COORD(PIECE,SEGLTH(J+1),XT1(I,J+2),XT2(I,J+1))NCI  510
      POSIT(I,II,2) = COORD(PIECE,SEGLTH(J+1),YT1(I,J+2),YT2(I,J+1))NCI  520
      POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI  530
      IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(PIECE,SEGLTH(J+1),ZT1(I,NCI  540
     $J+2),ZT2(I,J+1))                                                   NCI  550
      SEGLTH(J+1) = SEGLTH(J+1)-PIECE                                    NCI  560
      GO TO 70                                                           NCI  570
40    PIECE = SAMLTH-SEGLTH(J)                                           NCI  580
      GAMMA = PIECE/RADIUS                                               NCI  590
      ARCLTH(J) = ARCLTH(J)-PIECE                                        NCI  600
      II = II+1                                                          NCI  610
      ALFA1 = ATAN2((YT1(I,J+1)-YCENTR(I,J)),(XT1(I,J+1)-XCENTR(I,J)NCI  620
     $))                                                                 NCI  630
      IPLUS = 1                                                          NCI  640
      AA = ALFA1+ANGLE(I,J)/4.                                           NCI  650
      X22 = XCENTR(I,J)+RADIUS*COS(AA)                                   NCI  660
      Y22 = YCENTR(I,J)+RADIUS*SIN(AA)                                   NCI  670
      ADIS = SQRT((X22-XM(I,J+1))**2+(Y22-YM(I,J+1))**2)                 NCI  680
      BDIS = SQRT((XT1(I,J+1)-XM(I,J+1))**2+(YT1(I,J+1)-YM(I,J+1))**NCI  690
     $2)                                                                 NCI  700
      IF (BDIS.LT.ADIS) IPLUS = -1                                       NCI  710
      AA = ALFA1+IPLUS*GAMMA                                             NCI  720
      POSIT(I,II,1) = XCENTR(I,J)+RADIUS*COS(AA)                         NCI  730
      POSIT(I,II,2) = YCENTR(I,J)+RADIUS*SIN(AA)                         NCI  740
      POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI  750
      IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(PIECE,ARCLTH(J)+PIECE,ZTNCI  760
     $2(I,J+1),ZT1(I,J+1))                                               NCI  770
50    IF (ARCLTH(J).LT.SAMLTH) GO TO 60                                  NCI  780
      AA = AA+IPLUS*SAMLTH/RADIUS                                        NCI  790
      ARCLTH(J) = ARCLTH(J)-SAMLTH                                       NCI  800
      II = II+1                                                          NCI  810
      POSIT(I,II,1) = XCENTR(I,J)+RADIUS*COS(AA)                         NCI  820
      POSIT(I,II,2) = YCENTR(I,J)+RADIUS*SIN(AA)                         NCI  830
      POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI  840
      IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(SAMLTH,ARCLTH(J)+SAMLTH,NCI  850
     $ZT2(I,J+1),POSIT(I,II-1,3))                                        NCI  860
      GO TO 50                                                           NCI  870
60    PIECE = SAMLTH-ARCLTH(J)                                           NCI  880
      II = II+1                                                          NCI  890
      POSIT(I,II,2) = COORD(PIECE,SEGLTH(J+1),YT1(I,J+2),YT2(I,J+1))NCI  900
      POSIT(I,II,1) = COORD(PIECE,SEGLTH(J+1),XT1(I,J+2),XT2(I,J+1))NCI  910
      POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI  920
      IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(PIECE,SEGLTH(J+1),ZT1(I,NCI  930
     $J+2),ZT2(I,J+1))                                                   NCI  940
      SEGLTH(J+1) = SEGLTH(J+1)-PIECE                                    NCI  950
70    CONTINUE                                                           NCI  960
80  IF (SEGLTH(NSEG).LT.SAMLTH) GO TO 90                                 NCI  970
    II = II+1                                                            NCI  980
    POSIT(I,II,1) = COORD(SAMLTH,SEGLTH(NSEG),XT1(I,NSEG+1),POSIT(I,NCI  990
   $II-1,1))                                                             NCI 1000
    POSIT(I,II,2) = COORD(SAMLTH,SEGLTH(NSEG),YT1(I,NSEG+1),POSIT(I,NCI 1010
```

```
     $  II-1,2))                                                                NOI 1020
        POSIT(I,II,3) = PCSIT(I,II-1,3)-DLZPM*SAMLTH                            NOI 1030
        IF (IPRCF.EQ.1) PCSIT(I,II,3) = CCORD(SAMLTH,SEGLTH(NSEG),ZT1(I,NOI 1040
     $  NSEG+1),POSIT(I,II-1,3))                                                NOI 1050
        SEGLTH(NSEG) = SEGLTH(NSEG)-SAMLTH                                      NOI 1060
        GO TO 80                                                                NOI 1070
   90   NPOSIT(I) = II                                                          NOI 1080
        IF (II.GT.200) CALL ERRCP (8,'NO. OF TRAJ. SAMPLES .GT. 200,           NOI 1090
     $     RE-DIMENSION ARRAYS POSIT AND SPOSIT')                               NOI 1100
  100   CONTINUE                                                                NOI 1110
        NSAMP = IFIX(PERIOD*3600/SAMPT)                                         NOI 1120
        TVICLA = 0.                                                             NOI 1130
        DO 110 K = 1,NMAP                                                       NOI 1140
          AMBN = C.                                                             NOI 1150
C                                                                               NCI 1160
C ..............................................................                NOI 1170
C .                                                            .                NOI 1180
C .   CALC. ENERGY CONTRIB. BY AMBIENT NOISE                   .                NOI 1190
C .   CURRENTLY, AMBIENT LEVEL = 0. DB                         .                NOI 1200
C .                                                            .                NOI 1210
C ..............................................................                NOI 1220
C                                                                               NOI 1230
        ARRAY(K,4) = NSAMP*10.**(AMBN/10.)                                      NOI 1240
  110   CONTINUE                                                                NOI 1250
        DO 160 K = 1,NMAP                                                       NOI 1260
          BLKVIOL = 0.                                                          NOI 1270
          IF (ARRAY(K,3).LT.1) GO TO 150                                        NOI 1280
          ENERGY = 0.                                                           NOI 1290
          DO 140 I = 1,NTRAJ                                                    NOI 1300
            NPOSITI = NPOSIT(I)                                                 NOI 1310
            ENERGY = 0.                                                         NOI 1320
            DO 130 J = 1,NPOSITI                                                NOI 1330
              RANGE = SQRT((POSIT(I,J,1)-ARRAY(K,1))**2+(POSIT(I,J,2)-ARRAYNOI 1340
     $        Y(K,2))**2+(PCSIT(I,J,3))**2)                                      NOI 1350
              DO 120 L = 1,16                                                   NOI 1360
                IF (NAC(I,2*L).EQ.0.AND.NAC(I,2*L-1).EQ.0) GO TO 120            NOI 1370
                AL(L) = CNOIS(L,1)-CNOIS(L,2)*ALOG10(RANGE)-12.                 NOI 1380
                IF (AL(L).GE.ALMAX) BLKVIOL = BLKVIOL+AL(L)*(NAC(I,2*L-1)+NOI 1390
     $          NAC(I,2*L))                                                     NOI 1400
                ENERGY = ENERGY+10.**(AL(L)/10.)*NAC(I,2*L-1)+10.**(AL(L)/NOI 1410
     $          10.)*NAC(I,2*L)*10.                                             NOI 1420
  120           CONTINUE                                                      NOI 1430
  130         CONTINUE                                                        NOI 1440
            ARRAY(K,4) = ENERGY+ARRAY(K,4)                                      NOI 1450
  140       CONTINUE                                                          NOI 1460
  150     ARRAY(K,4) = 10.*ALOG10(ARRAY(K,4)/(86400/SAMPT))                   NOI 1470
          BLKVIOL = AMAX1(C.,BLKVIOL-ALMAX*MAXFLT)                            NOI 1480
          IF (ARRAY(K,10).EC.1.) TVICLA = TVICLA+BLKVIOL                      NOI 1490
  160   CONTINUE                                                              NOI 1500
        THRESH = THEIT1*TVICLA**2                                           NOI 1510
        RETURN                                                              NOI 1520
        END                                                                 NOI 1530
```

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE7,TAPE8,TAPE9,TAPE10)  MAI   10
C.....                                                                  MAI   20
C.                                                                      MAI   30
C.                    * * * *  C Y C L I C  * * * *                     MAI   40
C.                                                                      MAI   50
C.                  MINIMAL AIRCRAFT NOISE IMPACT PROGRAM               MAI   60
C.                                                                      MAI   70
C.                   WITH CYCLIC TRAJECTORY OPTIMIZATION                MAI   80
C.                                                                      MAI   90
C.              AUTHORS:  R. HELTON AND P. CHANG                        MAI  100
C.                        UNIVERSITY OF VIRGINIA                        MAI  110
C.                        CHARLOTTESVILLE, VA                           MAI  120
C.                                                                      MAI  130
C.                 REVISED:  MAY 26, 1981                               MAI  140
C.                                                                      MAI  150
C.....                                                                  MAI  160
C.                                                                      MAI  170
C.     THIS PROGRAM USES NONLINEAR OPTIMIZATION TECHNIQUES TO           MAI  180
C.     FIND THE SET OF AIRCRAFT TRAJECTORIES WITH THE MINIMUM           MAI  190
C.     NOISE IMPACT INDEX (NII) FOR A SPECIFIED MIX OF AIRCRAFT         MAI  200
C.     AND FOR THE PARTICULAR POPULATION DISTRIBUTION INPUT.            MAI  210
C.     THE EQUIVALENT NOISE CONCEPT IS USED, ALONG WITH AN APPROACH     MAI  220
C.    THAT PERFORMS THE OPTIMIZATION ON ONE TRAJECTORY AT A TIME        MAI  230
C.                                                                      MAI  240
C.                                                                      MAI  250
C.     THE SUBROUTINES USED ARE:                                        MAI  260
C.                                                                      MAI  270
C.        JPARAMS (NMACFTN PACKAGE ON NASA LRC PUBLIC LIB)              MAI  280
C.        COST,COST1,AACCS,MONIT,RESULT,CPCSCVP,ONEWTON,MPLY,TRNSPOS,   MAI  290
C.        FCRAD,GAFA,ERRCP,LINESCH,CHECK,CORNER,DYNAMIC,NII,NCISEL,     MAI  300
C.        PCVRY,KNORCEN,ADDR                                            MAI  310
C.                                                                      MAI  320
C.....                                                                  MAI  330
      EXTERNAL PCVRY                                                    MAI  340
      INTEGER ORD                                                       MAI  350
      LOGICAL ZERGLOB,NEWTRAJ,ZENABLE                                   MAI  360
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPCSIT(15,250MAI  370
     $,3),XO(15),XF(15),YO(15),YF(15),ZO(15),ZF(15),NPCSITS(15),PHO(573)MAI  380
     $,ISEG1,IPPCF,MAXIT,PENCPIT,AWEIT,TVIGLA,ALWP,PEOPLE,NTRCPT,TMPLON(MAI  390
     $573),CPD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)            MAI  400
      COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,ZWEIT1,GMAMIN(15),GMAMAX(15)MAI  410
     $,NSAPP                                                            MAI  420
      COMMON /THRESH/ ALMAX,MAXFLIT,TWEIT1,TWEIT2,NPLANE                MAI  430
      COMMON /CROSS/ XBEGIN,XFINAL,YOIS,CWEIT1,CWEIT2                   MAI  440
      COMMON /PPINT/ SXCENTR(15,12),SYCENTR(15,12),SXT1(15,12),SYT1(15,1MAI  450
     $2),SZT1(15,12),SXT2(15,12),SYT2(15,12),SZT2(15,12),SANGLE(15,12)  MAI  460
      COMMON /SAMPLE/ SAMPT,PERIOD,SAMLTH,NAC(15,32),CNOIS(16,2)        MAI  470
      COMMON /ENABLE/ ZENABLE                                           MAI  480
      DIMENSION XNOW(30,1), DELTAX(30), LABEL(8)                        MAI  490
      DIMENSION XPORT(15), YPORT(15), JN(30)                            MAI  500
      DIMENSION NDC830(15,2), NDC9SAM(15,2), NDC10(15,2), N707SAM(15,2),MAI  510
     $ N720(15,2), N72720C(15,2), N727SAM(15,2), N737100(15,2), N737SAM(MAI  520
     $15,2), N747200(15,2), NL1011(15,2), NA300(15,2), NBAC111(15,2), NVMAI  530
     $C10(15,2), NCV990(15,2), NSST(15,2), NAMAC(32)                    MAI  540
      EQUIVALENCE (NAC(1),NDC830(1)), (NAC(31),NDC9SAM(1)), (NAC(61),NDCMAI  550
     $10(1)), (NAC(91),N707SAM(1)), (NAC(121),N720(1)), (NAC(151),N72720MAI  560
     $0(1)), (NAC(181),N727SAM(1)), (NAC(211),N737100(1)), (NAC(241),N73MAI  570
     $7SAM(1)), (NAC(271),N747200(1)), (NAC(301),NL1011(1)), (NAC(331),NMAI  580
     $A300(1)), (NAC(361),NBAC111(1)), (NAC(391),NVC10(1)), (NAC(421),NCMAI  590
     $V990(1)), (NAC(451),NSST(1))                                      MAI  600
      DATA NAMAC/"DC-8-30   ","          ","DC-8 W/SAM","ENG       ","DCMAI  610
     $-10-10  ","          ","707 W/SAM ","ENG       ","720       ","   MAI  620
     $      ","727-200   ","          ","727 W/SAM ","ENG       ","737-MAI  630
     $100/20","0         ","737 W/SAM ","ENG       ","747-200   ","     MAI  640
```

```
     $     M,MIL1011     M,M          M,MA3CO      M,M          M,MBAC-11MAI  650
     $1    M,M           M,MVC-1C     M,M          M,MCV-990    M,M        MAI  660
     $    M,MCONCORDE SM,M5I       M/                                      MAI  670
      DATA NAC/480*0/                                                      MAI  680
      NAMELIST /ACMIX/ NOCE3C,NDC*SAM,NDC10,N707SAM,N72C,N727200,N727SAMMAI  690
     $,N737100,N737SAM,N747200,NL1011,NA3CO,NBAC111,NVC10,NCV990,NSST     MAI  700
      NAMELIST /OPCEP/ CPD,MAXITG,ZENABLE                                  MAI  710
C                                                                          MAI  720
C ***********************************************************************MAI  730
C .                                                                        . MAI  740
C .   INPUT PARAMETERS:                                                    . MAI  750
C .   ORDER = NAMELIST: CPD(I) = TRAJ. NO. WITH ORDINAL I                  . MAI  760
C .                     MAXITG = TOTAL NO. OF GLOBAL ITERATIONS            . MAI  770
C .                     ZENABLE = .TRUE., Z-COORDS OF CORNER PTS.          . MAI  780
C .                               ALLOWED TO VARY                          . MAI  790
C .                             = .FALSE., Z-COORDS HELD CONSTANT          . MAI  800
C .   IENTRY = 'FIXED' FOR FIXED ENTRY PTS.                                . MAI  810
C .          = 'VARIAB' FOR VARIABLE ENTRY PTS.                            . MAI  820
C .   ICNTINU = 0, INITIAL TRAJS ARE STRAIGHT LINES                        . MAI  830
C .           = 1, INITIAL TRAJS ARE USER INPUT                            . MAI  840
C .   SAMPT = TIME IN SECONDS BETWEEN NOISE LEVEL SAMPLES                  . MAI  850
C .   PERIOD = TIME IN HRS. OVER WHICH THE SIMULATION OCCURS               . MAI  860
C .   ACSPD = A/C SPEED IN METERS/SEC.                                     . MAI  870
C .   NTRAJ = NUMBER OF FLIGHT TRAJECTORIES                                . MAI  880
C .   NMAP = NUMBER OF POPULATION POINTS ON MAP                            . MAI  890
C .   NSEG = NUMBER OF LINE SEGMENTS ON EACH TRAJECTORY                    . MAI  900
C .   MAXIT = MAXIMUM NO. OF ITERATIONS ALLOWED                            . MAI  910
C .   IF ONLY RESULT OF INITIAL CONDITION IS NEEDED, MAXIT = 0             . MAI  920
C .   DELH = PERTURBATION (METERS) IN X,Y DIRECTIONS AT CORNER             . MAI  930
C .   POINTS FOR GRADIENT CALCULATION                                      . MAI  940
C .   STOPCHG = STOP CRITERION FOR SUCCESSIVE COST CHANGE                  . MAI  950
C .   XO,YO,ZO,XF,YF,ZF = STARTING AND FINAL POINTS OF TRAJECTORIES        . MAI  960
C .   XPORT,YPORT = AIRPORT LOCATION (METERS)                              . MAI  970
C .   XM,YM,ZM (I,J) = COORDS OF J-TH CORNER PT. ON I-TH TRAJ.             . MAI  980
C .   PWEIT1 = PENALTY WEIGHT ON DYN. CONSTR. ON TRAJS                     . MAI  990
C .   PWEIT2 = PENALTY WEIGHT ON CONSTR. ON FINAL ANGLE                    . MAI 1000
C .   RADIUS = MIN ALLOWED TURNING RADIUS (METERS)                         . MAI 1010
C .   XMIN,XMAX,YMIN,YMAX = BOUNDARIES OF AREA EXCLUDED                    . MAI 1020
C .                         FROM MAX. NOISE (THRESHOLD) CONSTR.           . MAI 1030
C .   ALMAX = MAX. ALLOWED A-LEVEL NOISE                                   . MAI 1040
C .   RATIO = FRACTION OF FLIGHTS ALLOWED TO VIOLATE MAX. A-LEVEL          . MAI 1050
C .   TWEIT1 = PENALTY WEIGHT ON MAX. NOISE AND NUMBER CONSTR.             . MAI 1060
C .   TWEIT2 = NOT USED NOW, BUT SOME VALUE MUST BE INPUT                  . MAI 1070
C .   XBEGIN,XFINAL = BOUNDARIES ON X-AXIS WHERE SEPARATING                . MAI 1080
C .                   CONSTR. IS IMPOSED                                   . MAI 1090
C .   YDIS = MIN. SEPARATING DISTANCE (Y-AXIS) BETWEEN TRAJS.              . MAI 1100
C .   CWEIT1,CWEIT2 = PENALTY WEIGHTS ON SEP. AND CROSSOVER                . MAI 1110
C .                   CONSTRS. RESPECTIVELY                                . MAI 1120
C .   IPROF = 0, NO PROFILE OPTIMIZATION                                   . MAI 1130
C .         1, PROFILE OPTIMIZATION OCCURS                                 . MAI 1140
C .   NOTE:  IPROF IS ALWAYS SET = 1 IN THIS PROGRAM                       . MAI 1150
C .   ZWEIT1 = PENALTY WEIGHT ON VERTICAL SLOPE CONSTR.                    . MAI 1160
C .   GMAMAX = MAX. ALLOWED VERTICAL ANGLE ON EACH SEG.                    . MAI 1170
C .   GMAMIN = MIN. ALLOWED VERTICAL ANGLE ON EACH SEG.                    . MAI 1180
C .   PENCRIT = MIN. SIGNIFICANT DYN. PENALTY LEVEL                        . MAI 1190
C .   AWEIT = SCALING ON PII                                               . MAI 1200
C .   ACMIX = NAMELIST: ACTYPE (I,1) = NO. OF ACTYPE ON TRAJ. I            . MAI 1210
C .                                    IN DAYTIME                          . MAI 1220
C .                     ACTYPE (I,2) = NO. OF ACTYPE ON TRAJ. I            . MAI 1230
C .                                    AT NIGHT                            . MAI 1240
C .                     GMAMIN (I) = MIN. FLIGHT PATH ANGLE (DEG) ON       . MAI 1250
C .                                  TRAJ. I                               . MAI 1260
C .                     GMAMAX (I) = MAX. FLIGHT PATH ANGLE (DEG) ON       . MAI 1270
C .                                  TRAJ. I                               . MAI 1280
```

```
C .                                                                    .  MAI 1290
C .....................................................................   MAI 1300
C                                                                         MAI 1310
      CALL RECOVR (RCVRY,778,0)                                           MAI 1320
      CALL DATE (DAT)                                                     MAI 1330
      ENCODE (10,9010,IDAT) DAT                                           MAI 1340
      CALL REMARK (IDAT)                                                  MAI 1350
      CALL JPARAMS (JN)                                                   MAI 1360
      ENCODE (7,9020,JOBNAME) JN(1)                                       MAI 1370
      CALL REMARK (JOBNAME)                                               MAI 1380
      WRITE (6,9130)                                                      MAI 1390
      WRITE (6,9030)                                                      MAI 1400
      WRITE (6,9040) JOBNAME,IDAT                                         MAI 1410
      NPLANE = 0                                                          MAI 1420
C                                                                         MAI 1430
C .....................................................................   MAI 1440
C .                                                                   .   MAI 1450
C .   INPUT ALL THE INFORMATION                                       .   MAI 1460
C .                                                                   .   MAI 1470
C .....................................................................   MAI 1480
C                                                                         MAI 1490
      READ (7,9150) (LABEL(I),I=1,8)                                      MAI 1500
      READ (7,ORDER)                                                      MAI 1510
      READ (7,9050) IENTRY                                                MAI 1520
      IF (IENTRY.NE."VARIA".AND.IENTRY.NE."FIXED") CALL ERROR (4,"ILLEGAMAI 1530
     $L ENTRY POINT TYPE, CHECK DATA")                                    MAI 1540
C                                                                         MAI 1550
C .....................................................................   MAI 1560
C .                                                                   .   MAI 1570
C .   ISEG1 = 1 :  FIXED ENTRY PTS., 1ST SEG. INCLUDED IN ALL CALCS   .   MAI 1580
C .   ISEG1 = 2 :  VARIABLE ENTRY PTS., 1ST SEG. EXCLUDED            .   MAI 1590
C .                                                                   .   MAI 1600
C .....................................................................   MAI 1610
C                                                                         MAI 1620
      ISEG1 = 1                                                           MAI 1630
      IF (IENTRY.EQ."VARIA") ISEG1 = 2                                    MAI 1640
      READ (7,*) ICNTINU,SAMPT,PERIOD,ACSPD                               MAI 1650
      READ (7,*) NTRAJ,NTRPT,NTRCNST,NMAP,NSEG,MAXIT                      MAI 1660
      READ (7,*) DELH,STOPCHG                                             MAI 1670
      SAMLTH = SAMPT*ACSPD                                                MAI 1680
      DO 30 I = 1,NTRAJ                                                   MAI 1690
        READ (7,*) XO(ORD(I)),YO(ORD(I)),ZO(ORD(I))                       MAI 1700
        READ (7,*) XF(ORD(I)),YF(ORD(I)),ZF(ORD(I))                       MAI 1710
        READ (7,*) XPORT(ORD(I)),YPORT(ORD(I))                            MAI 1720
C                                                                         MAI 1730
C .....................................................................   MAI 1740
C .                                                                   .   MAI 1750
C .   CALCULATE INITIAL CORNER POINTS                                 .   MAI 1760
C .                                                                   .   MAI 1770
C .....................................................................   MAI 1780
C                                                                         MAI 1790
        XM(ORD(I),1) = XO(ORD(I))                                         MAI 1800
        YM(ORD(I),1) = YO(ORD(I))                                         MAI 1810
        ZM(ORD(I),1) = ZO(ORD(I))                                         MAI 1820
        XM(ORD(I),NSEG+1) = XF(ORD(I))                                    MAI 1830
        YM(ORD(I),NSEG+1) = YF(ORD(I))                                    MAI 1840
        ZM(ORD(I),NSEG+1) = ZF(ORD(I))                                    MAI 1850
        XM(ORD(I),NSEG+2) = (XPORT(ORD(I))+XF(ORD(I)))/2.                 MAI 1860
        YM(ORD(I),NSEG+2) = (YPORT(ORD(I))+YF(ORD(I)))/2.                 MAI 1870
        ZM(ORD(I),NSEG+2) = ZF(ORD(I))/2.                                 MAI 1880
        IF (ICNTINU.EQ.1) GO TO 20                                        MAI 1890
        DO 10 J = 2,NSEG                                                  MAI 1900
          XM(I,J) = XO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(XF(I)-XO(I))          MAI 1910
          YM(I,J) = YO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(YF(I)-YO(I))          MAI 1920
```

```
      ZM(I,J) = ZO(I)+(FLOAT(J-1)/FLOAT(NSEG))*(ZF(I)-ZO(I))          MAI 1930
10    CONTINUE                                                        MAI 1940
      GO TO 30                                                        MAI 1950
20    READ (7,*) (XM(ORD(I),J),YM(ORD(I),J),ZM(ORD(I),J),J=2,NSEG)    MAI 1960
30    CONTINUE                                                        MAI 1970
      DO 40 I = 1,NMAP                                                MAI 1980
        DO 40 J = 4,6                                                 MAI 1990
        ARRAY(I,J) = 0.                                               MAI 2000
40    CONTINUE                                                        MAI 2010
      READ (7,*) PWEIT1,PWEIT2,RADIUS                                 MAI 2020
      READ (7,*) XMIN,XMAX,YMIN,YMAX                                  MAI 2030
      READ (7,*) ALMAX,RATIO,TWEIT1,TWEIT2                           MAI 2040
      READ (7,*) XBEGIN,XFINAL,YDIS,CWEIT1,CWEIT2                     MAI 2050
      READ (7,*) IPROF,ZWEIT1,PENCRIT,AWEIT                           MAI 2060
      READ (7,*) ((CNOIS(ORD(I),J),J=1,2),I=1,NTRAJ)                  MAI 2070
      READ (10,*) ((ARRAY(I,J),J=1,3),(ARRAY(I,J),J=6,9),RHO(I),I=1,NMAPMAI 2080
     $)                                                               MAI 2090
      READ (7,ACMIX)                                                  MAI 2100
      READ (7,*) (GMAMIN(ORD(I)),I=1,NTRAJ)                           MAI 2110
      READ (7,*) (GMAMAX(ORD(I)),I=1,NTRAJ)                           MAI 2120
      IPROF = 1                                                       MAI 2130
      DO 50 I = 1,NTRAJ                                               MAI 2140
        DO 50 L = 1,32                                                MAI 2150
        NPLANE = NPLANE+NAC(I,L)                                      MAI 2160
50    CONTINUE                                                        MAI 2170
      DO 60 I = 1,NMAP                                                MAI 2180
        ARRAY(I,10) = 1.                                              MAI 2190
        IF (XMIN.LE.ARRAY(I,1).AND.ARRAY(I,1).LE.XMAX.AND.YMIN.LE.ARRAY(MAI 2200
     $  I,2).AND.ARRAY(I,2).LE.YMAX.AND.ARRAY(I,3).NE.0.) ARRAY(I,10) = MAI 2210
     $  0.                                                            MAI 2220
60    CONTINUE                                                        MAI 2230
      MAXFLIT = IFIX(NPLANE*RATIO+0.5)                                MAI 2240
C                                                                     MAI 2250
C ***************************************************************** MAI 2260
C .                                                               .   MAI 2270
C .   PRINT INFORMATION INPUT                                     .   MAI 2280
C .                                                               .   MAI 2290
C ***************************************************************** MAI 2300
C                                                                     MAI 2310
      WRITE (6,9160) (LABEL(I),I=1,8)                                 MAI 2320
      WRITE (6,9140) (ORD(I),I=1,NTRAJ),MAXITG                        MAI 2330
      WRITE (6,9170) MAXIT,NTRAJ,NSEG,NMAP,DELH,STOPCHG               MAI 2340
      WRITE (6,9180) NTRAJ,NTROPT,NTRCNST                             MAI 2350
      WRITE (6,9060) SAMPT,PERIOD,ACSPD                               MAI 2360
      IF (IENTRY.EQ."FIXED") WRITE (6,9070)                           MAI 2370
      IF (IENTRY.EQ."VARIA") WRITE (6,9080)                           MAI 2380
      IF (IPROF.EQ.1) WRITE (6,9090)                                  MAI 2390
      IF (IPROF.EQ.0) WRITE (6,9100)                                  MAI 2400
      IF (ZENABLE) WRITE (6,9110)                                     MAI 2410
      IF (.NOT.ZENABLE) WRITE (6,9120)                                MAI 2420
      WRITE (6,9190) ALMAX,MAXFLIT,XMIN,XMAX,YMIN,YMAX                MAI 2430
      WRITE (6,9200) XBEGIN,XFINAL,YDIS                               MAI 2440
      WRITE (6,9210) RADIUS,PWEIT1,PWEIT2,ZWEIT1,TWEIT1,TWEIT2,CWEIT1,CWMAI 2450
     $EIT2,PENCRIT,AWEIT                                              MAI 2460
      WRITE (6,9220)                                                  MAI 2470
      WRITE (6,9230) (ORD(I),GMAMIN(ORD(I)),GMAMAX(ORD(I)),I=1,NTRAJ) MAI 2480
      DO 90 I = 1,NTRAJ                                               MAI 2490
        WRITE (6,9240) ORD(I),XO(ORD(I)),YO(ORD(I)),ZO(ORD(I)),XF(ORD(I))MAI 2500
     $  ,YF(ORD(I)),ZF(ORD(I))                                        MAI 2510
        NSEG1 = NSEG+1                                                MAI 2520
        NSEG2 = NSEG+2                                                MAI 2530
        DO 70 J = 1,NSEG2                                             MAI 2540
          WRITE (6,9260) J,XM(ORD(I),J),YM(ORD(I),J),ZM(ORD(I),J)     MAI 2550
70      CONTINUE                                                      MAI 2560
```

```
      WRITE (6,9270)                                                    MAI 2570
      DO 80 L = 1,31,2                                                  MAI 2580
        IF (NAC(I,L).NE.0.OR.NAC(I,L+1).NE.0) WRITE (6,9280) NAMAC(L),  MAI 2590
     $    NAMAC(L+1),NAC(I,L),NAC(I,L+1)                               MAI 2600
   80   CONTINUE                                                        MAI 2610
      WRITE (6,9250) XPORT(OPD(I)),YPORT(OPD(I))                        MAI 2620
   90 CONTINUE                                                          MAI 2630
      N = 2*(NSEG-1)                                                    MAI 2640
      NVAR = 2                                                          MAI 2650
      IF (IPROF.EQ.1) N = 3*(NSEG-1)                                    MAI 2660
      IF (IPROF.EQ.1) NVAR = 3                                          MAI 2670
      I = 1                                                             MAI 2680
      NSEG1 = NSEG-1                                                    MAI 2690
      DO 100 J = 1,NSEG1                                                MAI 2700
        DO 100 K = 1,NVAR                                               MAI 2710
        L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                               MAI 2720
        IF (K.EQ.1) XNOW(L,1) = XM(ORD(I),J+1)                          MAI 2730
        IF (K.EQ.2) XNOW(L,1) = YM(ORD(I),J+1)                          MAI 2740
        IF (K.EQ.3) XNOW(L,1) = ZM(OPD(I),J+1)                          MAI 2750
  100   CONTINUE                                                        MAI 2760
      DO 110 I = 1,N                                                    MAI 2770
        DELTAX(I) = DELH                                                MAI 2780
  110   CONTINUE                                                        MAI 2790
C                                                                       MAI 2800
C .....................................................................  MAI 2810
C .                                                                   .  MAI 2820
C .   START OPTIMIZATION                                              .  MAI 2830
C .                                                                   .  MAI 2840
C .....................................................................  MAI 2850
C                                                                       MAI 2860
      DO 120 K = 1,NMAP                                                 MAI 2870
        TMPLON(K) = 1.                                                  MAI 2880
  120   CONTINUE                                                        MAI 2890
      NDEC = 30                                                         MAI 2900
      ZEROLCB = .TRUE.                                                  MAI 2910
      CALL PNCROBN (MAXITG,STOPCHG,N,XNOW,DELTAX,NDEC)                  MAI 2920
      STOP                                                              MAI 2930
C                                                                       MAI 2940
 9010 FORMAT (A10)                                                      MAI 2950
 9020 FORMAT (A7)                                                       MAI 2960
 9030 FORMAT (///," NO AMBIENT NOISE INCLUDED",//)                      MAI 2970
 9040 FORMAT (1H1,9X,"............",/,10X,".",11X,".",/,10X,".",2X,A7,2MAI 2980
     $X,".",/,10X,".",11X,".",/,10X,".",A10,1X,".",/,10X,".",11X,".",/,1MAI 2990
     $0X,"............",/)                                             MAI 3000
 9050 FORMAT (A5)                                                       MAI 3010
 9060 FORMAT (5X,"FOR LON CALCULATION, SAMPLE TIME = ",F4.0," SEC.",/,5XMAI 3020
     $,"A/C LANDING PATTERN REPEATS EVERY ",F5.1," HRS.",/,5X,"A/C SPEEDMAI 3030
     $ = ",F6.2," M/S",///)                                             MAI 3040
 9070 FORMAT (/,4X,"FIXED ENTRY POINTS",/)                              MAI 3050
 9080 FORMAT (/,4X,"VARIABLE ENTRY POINTS",/)                           MAI 3060
 9090 FORMAT (/,4X,".. PROFILES OPTIMIZED ..",/)                        MAI 3070
 9100 FORMAT (/,4X,".. ONLY GROUND TRACKS ARE OPTIMIZED ..",/)          MAI 3080
 9110 FORMAT (/,4X,".. Z COORDINATES OF CORNER PTS. ALLOWED TO VARY",//)MAI 3090
 9120 FORMAT (/,4X,".. Z COORDINATES OF CORNER PTS. HELD CONSTANT",//)  MAI 3100
 9130 FORMAT (1H1,10H**********,///)                                    MAI 3110
 9140 FORMAT (/,5X,"INITIAL ORDER OF TRAJS: ",15(I2,2X),/,5X,"NO. OF GLCMAI 3120
     $BAL ITERATIONS = ",I2)                                            MAI 3130
 9150 FORMAT (8A10)                                                     MAI 3140
 9160 FORMAT (1X,8A10)                                                  MAI 3150
 9170 FORMAT (///,5X,"MAXIMUM ITERATION SET: ",I2,/,5X,"NUMBER OF TRA","MAI 3160
     $JECTORIES: ",I2,/,5X,"NUMBER OF SEGMENTS ON EACH TRAJECTORY: ",I1,MAI 3170
     $/,5X,"NUMBER OF POPULATION POINTS ON THE MAP: ",I4,/,5X,"PERTURB TMAI 3180
     $RAJECTORIES IN X AND Y DIRECTIONS ",F10.5," METERS FOR CALCULATINGMAI 3190
     $ GRADIENTS",/,5X,"EXIT CRITERION FOR ONEWTON: ",1PE16.9,/)        MAI 3200
```

```
 9180 FORMAT (/,5X,"TOTAL NUMBER OF TRAJECTORIES = ",I2,/,5X,"NUMBER OF MAI 3210
     $TRAJECTORIES TO BE OPTIMIZED = ",I2,/,5X,"NUMBER OF TRAJECTORIES FMAI 3220
     $ELD CONSTANT = ",I2,/)                                            MAI 3230
 9190 FORMAT (5X,"NO BLOCK MAY RECEIVE ",F5.1," DB (OR HIGHER) A-LEVEL",MAI 3240
     $" NOISE MORE THAN ",I4," TIMES A DAY",//,3X,"..THIS CONSTRAINT ","MAI 3250
     $EXCLUDES BLOCKS WITH CENTROIDS INSIDE THE RANGES: ",F8.1," TO ",FFMAI 3260
     $.1," (X) AND ",F8.1," TO ",F8.1," (Y)",//)                        MAI 3270
 9200 FORMAT (5X,40HWITHIN THE AREA OF X-COORDINATES BETWEEN,F8.1,5H ANDMAI 3280
     $ ,F8.1,41H, THE SEPARATING DISTANCE BETWEEN TRAJECT,5HORIES,/,5X,1MAI 3290
     $0HSHOULD BE AT LEAST,F6.1,7H METERS,//)                           MAI 3300
 9210 FORMAT (/,8X,"OTHER CONSTRAINT PARAMETERS",/,4X,"MINIMUM RADIUS","MAI 3310
     $ OF CURVATURE (IN X-Y PLANE PROJECTION) = ",1PE9.2,/,4X,"DYNAMIC PMAI 3320
     $ENALTY WEIGHTS:  PWEIT1 = ",E9.2,"  PWEIT2 = ",E9.2,"  ZWEIT1 = ",MAI 3330
     $E9.2,/,4X,"MAXIMUM NOISE LEVEL (THRESHOLD) ","PENALTY WEIGHTS:  TWMAI 3340
     $EIT1 = ",E9.2,"TWEIT2 = ",E9.2,/,4X,"SEPARATION PENALTY WEIGHTS:  MAI 3350
     $CWEIT1 = ",E9.2,"  CWEIT2 = ",E9.2,/,4X," PENCRIT = ",1PE9.2," AWEMAI 3360
     $IT = ",E9.2,//)                                                   MAI 3370
 9220 FORMAT (/,8X,"VERTICAL SLOPE CONSTRAINTS",/,8X,"TRAJ NO.",2X,"GMAMMAI 3380
     $IN  GMAMAX",//)                                                   MAI 3390
 9230 FORMAT (11X,I2,5X,F5.1,3X,F5.1)                                   MAI 3400
 9240 FORMAT (10X,15HFLIGHTPATH NO: ,I2,/,12X,13HINITIAL X,Y,Z,1X,23HCOCMAI 3410
     $RDINATES IN METERS: ,3(F9.1,3X),/,12X,11HFINAL X,Y,Z,1X,23HCOORDINMAI 3420
     $ATES IN METERS: ,3(F9.1,3X),/,12X,14HINITIAL CORNER,10H POINT POSIMAI 3430
     $TIONS: ,/,14X,10HCORNER NO.,6X,1HX,9X,1HY,9X,1HZ,/)               MAI 3440
 9250 FORMAT (10X,"RUNWAY LOCATION: ",F8.1,10X,F8.1,//)                 MAI 3450
 9260 FORMAT (18X,I2,6X,F9.1,3X,F9.1,3X,F9.1)                           MAI 3460
 9270 FORMAT (15X,"A/C MIX",T45,"NO. OF A/C",//,T40,"DAYTIME",3X,"NIGHTTMAI 3470
     $IME",//)                                                          MAI 3480
 9280 FORMAT (T20,2A10,T42,I3,T52,I3/)                                  MAI 3490
      END                                                               MAI 3500



      SUBROUTINE COST (IGRAD,TOTAL,ANII,PNALTY,CLOSE,THRESH)            COS   10
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25COS   20
     $,3),XO(15),XF(15),YO(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)COS   30
     $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTROPT,TMPLDN(COS   40
     $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)            COS   50
      INTEGER ORD                                                       COS   60
      LOGICAL ZERGLOB,NEWTRAJ                                           COS   70
      COMMON /CORNER/ ANGLE(15,12),POSIT(15,250,3),XCENTR(15,12),YCENTR(COS   80
     $15,12),ZCENTR(15,12),XT1(15,12),YT1(15,12),ZT1(15,12),XT2(15,12),YCOS   90
     $T2(15,12),ZT2(15,12),NPOSIT(15),DIS(15)                           COS  100
      COMMON /PRINT/ SXCENTR(15,12),SYCENTR(15,12),SXT1(15,12),SYT1(15,1COS  110
     $2),SZT1(15,12),SXT2(15,12),SYT2(15,12),SZT2(15,12),SANGLE(15,12)  COS  120
      TOTAL=0.                                                          COS  130
      ANII=TOTAL                                                        COS  140
      CLOSE=ANII                                                        COS  150
      THRESH=CLOSE                                                      COS  160
      PNALTY=THRESH                                                     COS  170
      CALL DYNAMID (PNALTY)                                             COS  180
C                                                                       COS  190
C .....................................................................COS  200
C .                                                                   .COS  210
C .   SUBTR. NOISE EFFECTS OF TRAJ. BEING OPT.                        .COS  220
C .   DURING PREV. CALL TO COST                                       .COS  230
C .                                                                   .COS  240
C .....................................................................COS  250
C                                                                       COS  260
      IF (.NOT.ZERGLOB) CALL ADDR (-1)                                  COS  270
      IF (PNALTY.LT.PENCRIT) GO TO 10                                   COS  280
      TOTAL = PNALTY                                                    COS  290
      IF (MAXIT.NE.0) RETURN                                            COS  300
```

```
10 CALL NOISEL (THRESH)                                                 COS  310
   CALL NII (ANII)                                                      COS  320
   IF (NTRAJ.EQ.1) GO TO 20                                             COS  330
   CALL CROSCVR (NPOSIT,POSIT,CLOSE)                                    COS  340
20 TOTAL = ANII+THRESH+CLOSE+PNALTY                                     COS  350
   IF (IGRAD.EQ.1) RETURN                                               COS  360
   DO 50 II = 1,NTRAJ                                                   COS  370
     I = ORD(II)                                                        COS  380
     SXT1(I,1) = XT1(I,1)                                               COS  390
     SXT2(I,1) = XT2(I,1)                                               COS  400
     SYT1(I,1) = YT1(I,1)                                               COS  410
     SYT2(I,1) = YT2(I,1)                                               COS  420
     SZT1(I,1) = ZT1(I,1)                                               COS  430
     SZT2(I,1) = ZT2(I,1)                                               COS  440
     DO 30 J = 2,NSEG                                                   COS  450
       SXCENTR(I,J-1) = XCENTR(I,J-1)                                   COS  460
       SYCENTR(I,J-1) = YCENTR(I,J-1)                                   COS  470
       SXT1(I,J) = XT1(I,J)                                             COS  480
       SYT1(I,J) = YT1(I,J)                                             COS  490
       SXT2(I,J) = XT2(I,J)                                             COS  500
       SYT2(I,J) = YT2(I,J)                                             COS  510
       SZT1(I,J) = ZT1(I,J)                                             COS  520
       SZT2(I,J) = ZT2(I,J)                                             COS  530
       SANGLE(I,J-1) = ANGLE(I,J-1)                                     COS  540
30     CONTINUE                                                         COS  550
     NPOSITS(I) = NPOSIT(I)                                             COS  560
     IPOST = NPOSITS(I)                                                 COS  570
     DO 40 J = 1,IPOST                                                  COS  580
       SPOSIT(I,J,1) = POSIT(I,J,1)                                     COS  590
       SPOSIT(I,J,2) = POSIT(I,J,2)                                     COS  600
       SPOSIT(I,J,3) = POSIT(I,J,3)                                     COS  610
40     CONTINUE                                                         COS  620
50   CONTINUE                                                           COS  630
   RETURN                                                               COS  640
   END                                                                  COS  650
```

```
   SUBROUTINE COST1 (IGRAD,N,F,X,ANII,PNALTY,CLOSE,THRESH,NDEC)         CO1   10
   COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,250CO1   20
  $,3),XO(15),XF(15),YO(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)CO1   30
  $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,PEOPLE,NTRCPT,TMPLDN(CO1   40
  $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)               CO1   50
   INTEGER ORD                                                          CO1   60
   LOGICAL ZERGLOB,NEWTRAJ                                              CO1   70
   DIMENSION X(NDEC,1)                                                  CO1   80
   NVAR = 2                                                             CO1   90
   IF (IPROF.EQ.1) NVAR = 3                                             CO1  100
   I = 1                                                                CO1  110
   NSEG1 = NSEG-1                                                       CO1  120
   DO 10 J = 1,NSEG1                                                    CO1  130
     DO 10 K = 1,NVAR                                                   CO1  140
     L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                                  CO1  150
     IF (K.EQ.1) XM(ORD(I),J+1) = X(L,1)                                CO1  160
     IF (K.EQ.2) YM(ORD(I),J+1) = X(L,1)                                CO1  170
     IF (K.EQ.3) ZM(ORD(I),J+1) = X(L,1)                                CO1  180
10   CONTINUE                                                           CO1  190
   CALL COST (IGRAD,F,ANII,PNALTY,CLOSE,THRESH)                         CO1  200
   RETURN                                                               CO1  210
   END                                                                  CO1  220
```

```
      FUNCTION AACOS (A,B,C)                                                AAC   10
      X = (A**2+B**2-C**2)/(2.*A*B)                                         AAC   20
      IF (ABS(X).LT.1.1) GO TO 10                                           AAC   30
      WRITE (6,9010) X,A,B,C                                                AAC   40
      STOP                                                                  AAC   50
   10 IF (X.GT.1.) X = 1.                                                   AAC   60
      IF (X.LT.-1.) X = -1.                                                 AAC   70
      AACOS = ACOS(X)                                                       AAC   80
      RETURN                                                                AAC   90
C                                                                           AAC  100
 9010 FORMAT (29H TROUBLE IN AACOS, X,A,B,C = ,4(1PE16.9,3X))               AAC  110
      END                                                                   AAC  120


      SUBROUTINE MCNIT (X,N,F,ANII,PNALTY,CLOSE,THRESH,NDEC)                MCN   10
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25CMCN   20
     $,3),XG(15),XF(15),YG(15),YF(15),ZG(15),ZF(15),NPOSITS(15),RHO(573)MCN   30
     $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTRGPT,TMPLONMCN   40
     $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)            MCN   50
      INTEGER ORD                                                           MCN   60
      LOGICAL ZERGLOB,NEWTRAJ                                               MCN   70
      COMMON /PRINT/ SXCENTR(15,12),SYCENTR(15,12),SXT1(15,12),SYT1(15,1MCN   80
     $2),SZT1(15,12),SXT2(15,12),SYT2(15,12),SZT2(15,12),SANGLE(15,12)  MCN   90
      DIMENSION X(NDEC,1)                                                   MCN  100
      PI = ATAN(1.)*4.                                                      MCN  110
      NVAR = 2                                                              MCN  120
      IF (IPROF.EQ.1) NVAR = 3                                              MCN  130
      I = 1                                                                 MCN  140
      NSEG1 = NSEG-1                                                        MCN  150
      DO 10 J = 1,NSEG1                                                     MCN  160
        DO 10 K = 1,NVAR                                                    MCN  170
        L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                                   MCN  180
        IF (K.EQ.1) XM(ORD(I),J+1) = X(L,1)                                 MCN  190
        IF (K.EQ.2) YM(ORD(I),J+1) = X(L,1)                                 MCN  200
        IF (K.EQ.3) ZM(ORD(I),J+1) = X(L,1)                                 MCN  210
   10   CONTINUE                                                            MCN  220
      WRITE (6,9010) IT,F,ANII,PNALTY,CLOSE,THRESH,TVIOLA,ALWP              MCN  230
      MAX = 1                                                               MCN  240
      IF (ZERGLOB) MAX = NTRAJ                                              MCN  250
      DO 30 I = 1,MAX                                                       MCN  260
        WRITE (6,9020) ORD(I),XM(ORD(I),1),YM(ORD(I),1),ZM(ORD(I),1)        MCN  270
        DO 20 J = 2,NSEG                                                    MCN  280
          WRITE (6,9030) XM(ORD(I),J),YM(ORD(I),J),ZM(ORD(I),J),SXCENTR(MCN  290
     $    ORD(I),J-1),SYCENTR(ORD(I),J-1),SXT1(ORD(I),J),SYT1(ORD(I),J),MCN  300
     $    SZT1(ORD(I),J),SXT2(ORD(I),J),SYT2(ORD(I),J),SZT2(ORD(I),J),SAMCN  310
     $    NGLE(ORD(I),J-1)*180./PI                                        MCN  320
   20     CONTINUE                                                         MCN  330
        WRITE (6,9040) XM(ORD(I),NSEG+1),YM(ORD(I),NSEG+1),ZM(ORD(I),NSEMCN  340
     $  G+1)                                                             MCN  350
        NPOSITI = NPOSITS(ORD(I))                                           MCN  360
   30   CONTINUE                                                            MCN  370
      WRITE (8,9050) (IFI)(ARRAY(K,3)),ARRAY(K,4),K=1,NMAP)                 MCN  380
      ZERGLOB = .FALSE.                                                     MCN  390
      RETURN                                                                MCN  400
C                                                                           MCN  410
 9010 FORMAT (1H1,9X,11HITERATION: ,I2,/,12X,12HTOTAL COST: ,1PE16.9,/,1MCN  420
     $2X,17HANNOYANCE (NII): ,1PE16.9,/,12X,31HPENALTY ON DYNAMIC CONSTRMCN  430
     $AINT: ,1PE16.9,/,12X,34HPENALTY ON SEPARATING CONSTRAINT: ,1PE16.9MCN  440
     $,/,12X,28HPENALTY ON THRESHOLD NOISE: ,1PE16.9,/,12X,"THRESH VIOLAMCN  450
     $TION = ",E16.9,/,12X,"LWP = ",E16.9,/)                            MCN  460
 9020 FORMAT (12X,"FLIGHT PATH NO: ",I2,/,T20,"CORNER PTS.",T40,"CENTER MCN  470
```

```
     $OF CIRCLE",T74,"TANGENTIAL PTS.",T110,"ANGLE (DEGREE)",/,14X,"(",FMCN  480
     $7.0,",",F7.0,",",F7.0,")")                                          MCN  490
9030 FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")",")",F7.0,",",F7.0,")",2MCN  500
     $("(",F7.0,",",F7.0,",",F7.0,")"),3X,F6.1)                           MCN  510
9040 FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")")                          MCN  520
9050 FORMAT (5X,I8,5X,F9.5)                                               MCN  530
     END                                                                  MCN  540



     SUBROUTINE RESULT (X,N,F,ANII,PNALTY,CLOSE,THRESH,NDEC)              RES   10
     COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,250RES   20
    $,3),XO(15),XF(15),YC(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)RES   30
    $,ISEC1,IPROF,MAXIT,FENCRIT,ALGIT,TVIOLA,ALWP,PEOPLE,NTROPT,TMPLON(RES   40
    $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)              RES   50
     INTEGER ORD                                                          RES   60
     LOGICAL ZERGLOR,NEWTRAJ                                              RES   70
     COMMON /PRINT/ SXCENTR(15,12),SYCENTR(15,12),SXT1(15,12),SYT1(15,1RES   80
    $2),SZT1(15,12),SYT2(15,12),SYT2(15,12),SZT2(15,12),SANGLE(15,12)    RES   90
     DIMENSION X(NDEC,1)                                                  RES  100
     PI = ATAN(1.)*4.                                                     RES  110
     NVAR = 2                                                             RES  120
     IF (IPROF.EQ.1) NVAR = 3                                             RES  130
     I = 1                                                                RES  140
     NSEG1 = NSEG-1                                                       RES  150
     DO 10 J = 1,NSEG1                                                    RES  160
       DO 10 K = 1,NVAR                                                   RES  170
       L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                                  RES  180
       IF (K.EQ.1) XM(ORD(I),J+1) = X(L,1)                                RES  190
       IF (K.EQ.2) YM(ORD(I),J+1) = X(L,1)                                RES  200
       IF (K.EQ.3) ZM(ORD(I),J+1) = X(L,1)                                RES  210
  10   CONTINUE                                                           RES  220
     WRITE (6,9010) IT,F,ANII,PNALTY,CLOSE,THRESH,TVIOLA,ALWP             RES  230
     DO 30 I = 1,NTRAJ                                                    RES  240
       WRITE (6,9020) ORD(I),XM(ORD(I),1),YM(ORD(I),1),ZM(ORD(I),1)       RES  250
       DO 20 J = 2,NSEG                                                   RES  260
         WRITE (6,9030) XM(ORD(I),J),YM(ORD(I),J),ZM(ORD(I),J),SXCENTR(RES  270
    $    ORD(I),J-1),SYCENTR(ORD(I),J-1),SXT1(ORD(I),J),SYT1(ORD(I),J),RES  280
    $    SZT1(ORD(I),J),SXT2(ORD(I),J),SYT2(ORD(I),J),SZT2(ORD(I),J),SARES  290
    $    NGLE(ORD(I),J-1)*180./PI                                        RES  300
  20     CONTINUE                                                        RES  310
       WRITE (6,9040) XM(ORD(I),NSEG+1),YM(ORD(I),NSEG+1),ZM(ORD(I),NSEGRES  320
    $  G+1)                                                            RES  330
       WRITE (6,9050)                                                  RES  340
       NPOSITI = NPOSITS(ORD(I))                                       RES  350
       WRITE (6,9060) ((SPOSIT(ORD(I),J,K),K=1,3),J=1,NPOSITI)         RES  360
  30   CONTINUE                                                        RES  370
     RETURN                                                           RES  380
C                                                                     RES  390
 9010 FORMAT (1H1,9X,11HITERATION: ,I2,/,12X,12HTOTAL COST: ,1PE16.9,/,1RES  400
     $2X,17HANNOYANCE (NII): ,1PE16.9,/,12X,31HPENALTY ON DYNAMIC CONSTRRES  410
     $AINT: ,1PE16.9,/,12X,24HPENALTY ON SEPARATING CONSTRAINT: ,1PE16.9RES  420
     $,/,12X,28HPENALTY ON THRESHOLD NOISE: ,1PE16.9,/,12X,"THRESH VIOLARES  430
     $TION = ",E16.9,/,12X,"LWP = ",E16.9,/)                          RES  440
 9020 FORMAT (12X,"FLIGHT PATH NO: ",I2,/,T20,"CORNER PTS.",T40,"CENTER RES  450
     $OF CIRCLE",T74,"TANGENTIAL PTS.",T110,"ANGLE (DEGREE)",/,14X,"(",FRES  460
     $7.0,",",F7.0,",",F7.0,")")                                      RES  470
 9030 FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")",")",F7.0,",",F7.0,")",2RES  480
     $("(",F7.0,",",F7.0,",",F7.0,")"),3X,F6.1)                       RES  490
 9040 FORMAT (14X,"(",F7.0,",",F7.0,",",F7.0,")")                     RES  500
 9050 FORMAT (//,12X,36HFLIGHT TRAJECTORY X, Y, Z, IN METERS,/,19X,1HX,1RES  510
     $4X,1HY,14X,1HZ,/)                                               RES  520
 9060 FORMAT (14X,1PE10.3,5X,E10.3,5X,E10.3,5X)                       RES  530
```

```
      END                                                                PES  540



      SUBROUTINE CROSOVR (I,A,P)                                         CRO   10
C                                                                        CRO   20
C ......................................................................CRO   30
C .                                                                    . CRO   40
C .   THIS IS A DUMMY ROUTINE, REPLACING THE ORIGINAL CROSOVR,         . CRO   50
C .   WHICH DOS NCT WORK FOR TRAJS THAT DONT SHARE A COMMON            . CRO   60
C .   SEGMENT OF THE X-AXIS                                            . CRO   70
C .                                                                    . CRO   80
C ......................................................................CRO   90
C                                                                        CRO  100
      RETURN                                                             CRO  110
      END                                                                CRO  120



      SUBROUTINE ONEWTON (N,XNOW,DELTAX,STOPCHG,NDEC)                    ONE   10
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25ONE   20
     $,3),XO(15),XF(15),YO(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)ONE   30
     $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVICLA,ALWP,PEOPLE,NTROPT,TMPLON(ONE   40
     $573),ORD(15),ITG,ICPT,IT,ZEROGLOB,NEWTRAJ,ARRAY(573,10)           ONE   50
C                                                                        ONE   60
C ......................................................................ONE   70
C .                                                                    . ONE   80
C .   THIS OPTIMIZATION EMPLOYS SELF-SCALING, RESTARTING,              . ONE   90
C .   QUASI-NEWTON METHOD.                                             . ONE  100
C .   REFERENCE:  D.G. LUENBERGER INTRO. TO LINEAR AND NONLINEAR       . ONE  110
C .   PROGRAMMING; P.204, SEC.9.5                                      . ONE  120
C .   MAXIT:  MAXIMUM NUMBER OF ITERATIONS ALLOWED                     . ONE  130
C .   STOPCHG:  STOP IF PERCENTAGE CHANGE IN SUCCESSIVE COSTS IS       . ONE  140
C .   LESS THAN THIS VALUE                                             . ONE  150
C .   N:  DIMENSION OF THE UNKNOWN X                                   . ONE  160
C .   XNOW:  PRESENT OR INITIAL VALUE OF UNKNOWN X                     . ONE  170
C .                                                                    . ONE  180
C ......................................................................ONE  190
C                                                                        ONE  200
      DIMENSION XNOW(NDEC,1), DELTAX(NDEC)                               ONE  210
      DIMENSION GNOW(30,1), GNEXT(30,1), P(30,1), Q(30,1), PQ(1,1)       ONE  220
      DIMENSION QSQ(1,1), PP(30,30), SQQS(30,30), S(30,30)               ONE  230
      DIMENSION XTEMP(30,1), PT(1,30), QS(1,30), SQ(1,30), QT(1,30)      ONE  240
      DIMENSION SGG(30,30)                                               ONE  250
      DIMENSION D(30,1)                                                  ONE  260
      IT = 0                                                             ONE  270
      CALL COST1 (0,N,FNOW,XNOW,ANII,PNALTY,CLOSE,THRESH,NDEC)           ONE  280
      IF (PNALTY.GE.PENCRIT) GO TO 10                                    ONE  290
      CALL MONIT (XNOW,N,FNOW,ANII,PNALTY,CLOSE,THRESH,NDEC)             ONE  300
      IF (MAXIT.EQ.0) RETURN                                             ONE  310
      GO TO 20                                                           ONE  320
   10 WRITE (6,9030) PNALTY,ANII                                         ONE  330
      CALL CORNER (XNOW,N,NDEC)                                          ONE  340
      IF (MAXIT.EQ.0) RETURN                                             ONE  350
C                                                                        ONE  360
C ......................................................................ONE  370
C .                                                                    . ONE  380
C .   STEP 1:  SET S = IDENTITY MATRIX AND CALCULATE GRADIENT G        . ONE  390
C .                                                                    . ONE  400
C ......................................................................ONE  410
C                                                                        ONE  420
   20 DO 30 I = 1,N                                                      ONE  430
```

```
      DO 30 J = 1,N                                                    ONE  440
        S(I,J) = 0.                                                    ONE  450
        IF (I.EQ.J) S(I,J) = 1.                                        ONE  460
   30   CONTINUE                                                       ONE  470
        CALL FGRAD (N,FNOW,XNOW,GNOW,DELTAX,NDEC)                      ONE  480
C                                                                      ONE  490
C ..................................................................   ONE  500
C .                                                                .   ONE  510
C .   STEP 2:  SET D = -SG                                          .   ONE  520
C .                                                                .   ONE  530
C ..................................................................   ONE  540
C                                                                      ONE  550
   40 CALL MPLY (N,N,1,S,GNOW,D,NDEC,NDEC,1)                          ONE  560
      DO 50 I = 1,N                                                    ONE  570
        D(I,1) = -D(I,1)                                               ONE  580
   50   CONTINUE                                                       ONE  590
C                                                                      ONE  600
C ..................................................................   ONE  610
C .                                                                .   ONE  620
C .   STEP 3:  MOST IMPORTANT!                                       .   ONE  630
C .     LINE SEARCH ALONG D TO FIND AFA THAT SATISFIES PG>0          .   ONE  640
C .     IF COST FUNCTION IS ANALYTIC, SLIGHT MODIFICATION IS         .   ONE  650
C .     NEEDED (SIMPLIFICATION)(OPTIONAL)                            .   ONE  660
C .                                                                .   ONE  670
C ..................................................................   ONE  680
C                                                                      ONE  690
      K = 0                                                            ONE  700
   60 K = K+1                                                          ONE  710
      CALL LINESCH (K,N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,CLOSE,ONE  720
     $THRESH,NDEC), RETURNS (220)                                      ONE  730
      IF (IT+1.LT.MAXIT) GO TO 70                                      ONE  740
      IT = IT+1                                                        ONE  750
      WRITE (6,9050)                                                   ONE  760
      IF (PNALTY.GE.PENCRIT) GO TO 150                                 ONE  770
      IF (FSMALL.LE.FNOW) CALL MONIT (XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THONE  780
     $RESH,NDEC)                                                       ONE  790
      RETURN                                                           ONE  800
   70 CALL FGRAD (N,FSMALL,XTEMP,GNEXT,DELTAX,NDEC)                    ONE  810
      DO 80 I = 1,N                                                    ONE  820
        Q(I,1) = GNEXT(I,1)-GNOW(I,1)                                  ONE  830
        P(I,1) = AFA2*D(I,1)                                           ONE  840
   80   CONTINUE                                                       ONE  850
      CALL TRNSPOS (NDEC,1,P,PT)                                       ONE  860
      CALL MPLY (1,N,1,PT,Q,PQ,1,NDEC,1)                               ONE  870
      IF (PQ(1,1).GT.0.) GO TO 110                                     ONE  880
      IF (K.LE.1) GO TO 60                                             ONE  890
      WRITE (6,9010)                                                   ONE  900
      RETURN                                                           ONE  910
   90 FNOW = FSMALL                                                    ONE  920
      DO 100 I = 1,N                                                   ONE  930
        XNOW(I,1) = XTEMP(I,1)                                         ONE  940
  100   CONTINUE                                                       ONE  950
      GO TO 60                                                         ONE  960
  110 DO 120 I = 1,N                                                   ONE  970
        XNOW(I,1) = XTEMP(I,1)                                         ONE  980
  120   CONTINUE                                                       ONE  990
      IT = IT+1                                                        ONE 1000
      PRCNT = ABS((FSMALL-FNOW)/FNOW)                                  ONE 1010
      IF (PRCNT.GE.STOPCHG) GO TO 140                                  ONE 1020
      WRITE (6,9040) PRCNT,STOPCHG                                     ONE 1030
      IF (PNALTY.GE.PENCRIT) GO TO 130                                 ONE 1040
      CALL MONIT (XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH,NDEC)         ONE 1050
C                                                                      ONE 1060
      RETURN                                                           ONE 1070
```

```
  130 WRITE (6,9020) PNALTY                                                 ONE 1080
      CALL CORNER (XNOW,N,NDEC)                                             ONE 1090
      RETURN                                                                ONE 1100
  140 IF (IT.LT.MAXIT) GO TO 160                                            ONE 1110
      WRITE (6,9050)                                                        ONE 1120
      IF (PNALTY.GE.PENCRIT) GO TO 150                                      ONE 1130
      CALL MONIT (XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH,NDEC)              ONE 1140
C                                                                           ONE 1150
      RETURN                                                                ONE 1160
  150 WRITE (6,9020) PNALTY                                                 ONE 1170
      CALL CORNER (XNOW,N,NDEC)                                             ONE 1180
      RETURN                                                                ONE 1190
  160 IF (PNALTY.GE.PENCRIT) GO TO 170                                      ONE 1200
      CALL MONIT (XNOW,N,FSMALL,ANII,PNALTY,CLOSE,THRESH,NDEC)              ONE 1210
      GO TO 180                                                             ONE 1220
  170 WRITE (6,9030) PNALTY,ANII                                            ONE 1230
      CALL CORNER (XNOW,N,NDEC)                                             ONE 1240
  180 CONTINUE                                                              ONE 1250
C                                                                           ONE 1260
C ..........................................................                ONE 1270
C .                                                        .                ONE 1280
C .   STEP 4:  IF "IT" IS INTEGER MULTIPLE OF N GO TO STEP 1;              . ONE 1290
C .         IF NOT, UPDATE S                               .                ONE 1300
C .     IS "IT" INTEGER MULTIPLE OF N?                     .                ONE 1310
C .                                                        .                ONE 1320
C ..........................................................                ONE 1330
C                                                                           ONE 1340
      IF ((FLOAT(IT)/FLOAT(N)).NE.FLOAT(IT/N)) GO TO 190                    ONE 1350
      FNOW = FSMALL                                                         ONE 1360
      GO TO 20                                                              ONE 1370
C                                                                           ONE 1380
C ..........................................................                ONE 1390
C .                                                        .                ONE 1400
C .     UPDATE MATRIX S; GO TO STEP 2                      .                ONE 1410
C .                                                        .                ONE 1420
C ..........................................................                ONE 1430
C                                                                           ONE 1440
  190 CALL MPLY (N,N,1,S,G,SG,NDEC,NDEC,1)                                  ONE 1450
      CALL TRNSPOS (NDEC,1,G,GT)                                            ONE 1460
      CALL MPLY (N,1,N,SG,GT,SGG,NDEC,1,NDEC)                               ONE 1470
      CALL MPLY (N,N,N,SGG,S,SGGS,NDEC,NDEC,NDEC)                           ONE 1480
      CALL MPLY (1,N,N,GT,S,GS,1,NDEC,NDEC)                                 ONE 1490
      CALL MPLY (1,N,1,GS,G,GSG,1,NDEC,1)                                   ONE 1500
      CALL MPLY (N,1,N,P,PT,PP,NDEC,1,NDEC)                                 ONE 1510
      DO 200 I = 1,N                                                        ONE 1520
        DO 200 J = 1,N                                                      ONE 1530
        S(I,J) = (S(I,J)-SGGS(I,J)/GSG(1,1))+(PG(1,1)/GSG(1,1))+PP(I,J)/ONE 1540
     $  PG(1,1)                                                             ONE 1550
  200   CONTINUE                                                            ONE 1560
      FNOW = FSMALL                                                         ONE 1570
      DO 210 I = 1,N                                                        ONE 1580
        GNOW(I,1) = GNEXT(I,1)                                              ONE 1590
  210   CONTINUE                                                            ONE 1600
      GO TO 40                                                              ONE 1610
  220 WRITE (6,9060) CFO(1)                                                 ONE 1620
      RETURN                                                                ONE 1630
C                                                                           ONE 1640
 9010 FORMAT (/,10X,"AFTER 2 CALLS TO LINESCH, CANT FIND TRAJ. WITH"," LONE 1650
     $OWER COST.  RETURN TO PROBN.")                                        ONE 1660
 9020 FORMAT (5X,40HIN RESULT: DYNAMIC CONSTRAINTS VIOLATION,/,10X,"PENONE 1670
     $ALTY =",1PE16.9)                                                      ONE 1680
 9030 FORMAT (5X,39HIN MONIT: DYNAMIC CONSTRAINTS VIOLATION,/,10X,"PENAONE 1690
     $LTY =",1PE16.9,/,10X,"ANII = ",E16.9," (IF = 0., THEN ANII HAS","NOONE 1700
     $T BEEN COMPUTED IN THIS ITERATION, DUE TO LARGE PNALTY")           ONE 1710
```

```
 9040 FORMAT (2X,"RELATIVE CHANGE IN SUCCESSIVE COSTS ",1PE10.3,", LESS"ONE 1720
     $," THAN STOP CRITERION ",1PE10.3)                                 ONE 1730
 9050 FORMAT (2X,29HMAXIMUM ITERATION SET REACHED)                      ONE 1740
 9060 FORMAT (/," ..... OPTIMIZATION FAILS FOR TRAJ. ",I2," ON THIS"," GONE 1750
     $LOBAL ITERATION",//)                                              ONE 1760
      END                                                               ONE 1770



      SUBROUTINE MPLY (L,M,N,A,B,C,LDEC,MDEC,NDEC)                      MPL   10
C                                                                       MPL   20
C ......................................................................MPL   30
C .                                                                    .MPL   40
C .   CALCULATE MATRIX MULTIPLICATION C = AB                           .MPL   50
C .                                                                    .MPL   60
C ......................................................................MPL   70
C                                                                       MPL   80
      DIMENSION A(LDEC,MDEC), B(MDEC,NDEC), C(LDEC,NDEC)               MPL   90
      DO 10 I = 1,L                                                     MPL  100
        DO 10 J = 1,N                                                   MPL  110
        C(I,J) = 0.                                                     MPL  120
        DO 10 K = 1,M                                                   MPL  130
        C(I,J) = C(I,J)+A(I,K)*B(K,J)                                   MPL  140
   10   CONTINUE                                                        MPL  150
      RETURN                                                            MPL  160
      END                                                               MPL  170



      SUBROUTINE TRNSPOS (M,N,A,B)                                      TRN   10
C                                                                       TRN   20
C ......................................................................TRN   30
C .                                                                    .TRN   40
C .   TRANSPOSE OF MATRIX A IS RETURNED IN MATRIX B                    .TRN   50
C .                                                                    .TRN   60
C ......................................................................TRN   70
C                                                                       TRN   80
      DIMENSION A(M,N), B(N,M)                                          TRN   90
      DO 10 I = 1,M                                                     TRN  100
        DO 10 J = 1,N                                                   TRN  110
        B(J,I) = A(I,J)                                                 TRN  120
   10   CONTINUE                                                        TRN  130
      RETURN                                                            TRN  140
      END                                                               TRN  150



      SUBROUTINE FGRAD (N,F,X,G,DELTAX,NDEC)                            FGR   10
C                                                                       FGR   20
C ......................................................................FGR   30
C .                                                                    .FGR   40
C .   CALCULATE GRADIENT OF COST F WITH RESPECT TO UNKNOWN X           .FGR   50
C .                                                                    .FGR   60
C ......................................................................FGR   70
C                                                                       FGR   80
      COMMON /ENABLE/ ZENABLE                                           FGR   90
      LOGICAL ZENABLE                                                   FGR  100
      DIMENSION X(NDEC,1), G(NDEC,1), DELTAX(NDEC,1)                    FGR  110
      DO 10 I = 1,N                                                     FGR  120
        G(I,1) = 0.                                                     FGR  130
        IF (.NOT.ZENABLE.AND.MOD(I,3).EQ.0) GO TO 10                    FGR  140
```

```
      X(I,1) = X(I,1)+DELTAX(I)                                          FGR  150
      CALL COST1 (1,N,FF,X,ANII,PNALTY,CLOSE,THRESH,NDEC)                FGR  160
      G(I,1) = (FF-F)/DELTAX(I)                                          FGR  170
      X(I,1) = X(I,1)-DELTAX(I)                                          FGR  180
   10 CONTINUE                                                           FGR  190
      RETURN                                                             FGR  200
      END                                                                FGR  210


      SUBROUTINE GAFA (N,F,DAFA,G,X,D,NDEC)                              GAF   10
C                                                                        GAF   20
C ***********************************************************************GAF   30
C *                                                                     *GAF   40
C *   CALCULATE GRADIENT OF F WITH RESPECT TO AFA                       *GAF   50
C *                                                                     *GAF   60
C ***********************************************************************GAF   70
C                                                                        GAF   80
      DIMENSION X(NDEC,1), D(NDEC,1), XTEMP(30,1)                        GAF   90
      DO 10 I = 1,N                                                      GAF  100
        XTEMP(I,1) = X(I,1)+DAFA*D(I,1)                                  GAF  110
   10 CONTINUE                                                           GAF  120
      CALL COST1 (1,N,FF,XTEMP,ANII,PNALTY,CLOSE,THRESH,NDEC)            GAF  130
      G = (FF-F)/DAFA                                                    GAF  140
      RETURN                                                             GAF  150
      END                                                                GAF  160


      SUBROUTINE ERROR (K,MESSAGE)                                       ERR   10
      DIMENSION MESSAGE(12), ITEXT(12)                                   ERR   20
      DATA ITEXT/12*"          "/                                        ERR   30
      DO 10 I = 1,K                                                      ERR   40
   10   ITEXT(I) = MESSAGE(I)                                           ERR   50
      WRITE (6,9010) (ITEXT(I),I=1,K)                                    ERR   60
      STOP                                                               ERR   70
C                                                                        ERR   80
 9010 FORMAT (//,1X,"ERROR: ",12A10,//)                                  ERR   90
      END                                                                ERR  100


      SUBROUTINE LINESCH (K,N,FNOW,XNOW,D,AFA2,XTEMP,FSMALL,ANII,PNALTY,LIN   10
     $CLOSE,THRESH,NDEC), RETURNS (KABORT)                               LIN   20
      DIMENSION D(NDEC,1), XTEMP(NDEC,1), XNOW(NDEC,1)                   LIN   30
C                                                                        LIN   40
C ***********************************************************************LIN   50
C *                                                                     *LIN   60
C *   CUBIC FIT BY INITIALLY LETTING AFA = 0 AND DAFA = 5.              *LIN   70
C *                                                                     *LIN   80
C ***********************************************************************LIN   90
C                                                                        LIN  100
      DNORM = 0.                                                         LIN  110
      DO 10 I = 1,N                                                      LIN  120
        DNORM = DNORM+D(I,1)**2                                          LIN  130
   10   CONTINUE                                                        LIN  140
      DNORM = SQRT(DNORM)                                                LIN  150
      IF (DNORM.EQ.0.) CALL CHECK (1,N,FNOW,XNOW,D,AFA0,FAFA0,GAFA0,AFA1LIN  160
     $,FAFA1,GAFA1,U1,U01,U2,NDEC)                                       LIN  170
C                                                                        LIN  180
C ***********************************************************************LIN  190
```

```
C .                                                                    . LIN  200
C .   NORMALIZE D                                                      . LIN  210
C .                                                                    . LIN  220
C ..................................................................... LIN  230
C                                                                        LIN  240
      DO 20 I = 1,N                                                      LIN  250
        D(I,1) = D(I,1)/DNORM                                            LIN  260
   20   CONTINUE                                                         LIN  270
      DAFA = 5.                                                          LIN  280
      AFA0 = 0.                                                          LIN  290
      FAFA0 = FNOW                                                       LIN  300
      KK = 1                                                             LIN  310
      CALL GAFA (N,FAFA0,DAFA,GAFA0,XNOW,D,NDEC)                         LIN  320
      IF (GAFA0.EQ.0) CALL CHECK (1,N,FNOW,XNOW,D,AFA0,FAFA0,GAFA0,AFA1,LIN  330
     $FAFA1,GAFA1,U1,LL1,L2,NDEC), RETURNS (90)                          LIN  340
      AFA1 = AMIN1(1000.,-FAFA0/GAFA0)                                   LIN  350
      IF (-FAFA0/GAFA0.LT.0) AFA1 = AMAX1(-1000.,-FAFA0/GAFA0)           LIN  360
   30 DO 40 I = 1,N                                                      LIN  370
        XTEMP(I,1) = XNOW(I,1)+AFA1*D(I,1)                               LIN  380
   40   CONTINUE                                                         LIN  390
      CALL COST1 (1,N,FAFA1,XTEMP,ANII,PNALTY,CLOSE,THRESH,NDEC)         LIN  400
      IF (KK.LT.4) GO TO 50                                              LIN  410
      WRITE (6,9040)                                                     LIN  420
      RETURN                                                             LIN  430
   50 CALL GAFA (N,FAFA1,DAFA,GAFA1,XTEMP,D,NDEC)                        LIN  440
      IF (ABS((AFA0-AFA1)/AFA1).LT..001) CALL CHECK (2,N,FNOW,XNOW,D,AFALIN  450
     $0,FAFA0,GAFA0,AFA1,FAFA1,GAFA1,U1,UU1,U2,NDEC), RETURNS (90)       LIN  460
      U1 = GAFA0+GAFA1-3.*(FAFA0-FAFA1)/(AFA0-AFA1)                      LIN  470
      UU1 = U1**2-GAFA0*GAFA1                                            LIN  480
      IF (UU1.LE.0.) U2 = 0.                                             LIN  490
      IF (UU1.GT.0.) U2 = SQRT(UU1)                                      LIN  500
      A0 = AMIN1(AFA0,AFA1)                                              LIN  510
      A1 = AMAX1(AFA0,AFA1)                                              LIN  520
      G0 = GAFA0                                                         LIN  530
      F0 = FAFA0                                                         LIN  540
      G1 = GAFA1                                                         LIN  550
      F1 = FAFA1                                                         LIN  560
      IF (AFA0.LT.AFA1) GO TO 60                                         LIN  570
      G0 = GAFA1                                                         LIN  580
      F0 = FAFA1                                                         LIN  590
      G1 = GAFA0                                                         LIN  600
      F1 = FAFA0                                                         LIN  610
   60 IF (G1-G0+2.*U2.EQ.0.) CALL CHECK (3,N,FNOW,XNOW,D,A0,F0,G0,A1,F1,LIN  620
     $G1,U1,LL1,U2,NDEC), RETURNS (90)                                   LIN  630
      AFA2 = A1-(A1-A0)*(G1+L2-U1)/(G1-G0+2.*U2)                         LIN  640
C                                                                        LIN  650
C ..................................................................... LIN  660
C .                                                                    . LIN  670
C .   RESTRICT MAXIMUM AFA2 TO 1.E4                                    . LIN  680
C .                                                                    . LIN  690
C ..................................................................... LIN  700
C                                                                        LIN  710
      AFA2 = AMIN1(AFA2,1.E4)                                            LIN  720
      IF (AFA2.GT.0.) AFA2 = AMIN1(AFA2,1.E4)                            LIN  730
      IF (AFA2.LT.0.) AFA2 = AMAX1(AFA2,-1.E4)                           LIN  740
      DO 70 I = 1,N                                                      LIN  750
        XTEMP(I,1) = XNOW(I,1)+AFA2*D(I,1)                               LIN  760
   70   CONTINUE                                                         LIN  770
      WRITE (6,9010) A0,A1,AFA2,DNORM,F0,F1,G0,G1                        LIN  780
      WRITE (6,9020) (D(I,1),I=1,N)                                      LIN  790
      CALL COST1 (0,N,FSMALL,XTEMP,ANII,PNALTY,CLOSE,THRESH,NDEC)        LIN  800
      WRITE (6,9020) FSMALL                                              LIN  810
      IF (FSMALL.GE.FNOW) GO TO 80                                       LIN  820
      RETURN                                                             LIN  830
```

```
   80 AFA1 = AFA1/2.                                                    LIN  840
      KK = KK+1                                                         LIN  850
      GO TO 30                                                          LIN  860
   90 RETURN KABORT                                                     LIN  870
C                                                                       LIN  880
 9010 FORMAT (15X,"A0 = ",1PE13.6,5X,"A1 = ",E13.6,5X,"AFA2 = ",E13.6,5XLIN  890
     $,"DNORM = ",E13.6,/,15X,"F0 = ",E13.6,5X,"F1 = ",E13.6,/,15X,"G0 =LIN  900
     $ ",E13.6,5X,"G1 = ",E13.6,/)                                      LIN  910
 9020 FORMAT (15X,"FSMALL = ",1PE13.6,/)                                LIN  920
 9030 FORMAT (5X,"D = ",10(1X,1PE9.2,2X))                               LIN  930
 9040 FORMAT(//,10X,"LINESCH CANT FIND LOWER COST AFTER 3 CUBIC FITS")  LIN  940
      END                                                               LIN  950
```

```
      SUBROUTINE CHECK (ICHECK,N,FNOW,XNOW,D,A0,F0,G0,A1,F1,G1,U1,UU1,U2CHE   10
     $,NDEC), RETURNS (KABORT)                                          CHE   20
      DIMENSION XNOW(NDEC,1), D(NDEC,1)                                 CHE   30
      WRITE (6,10) FNOW,N,(I,XNOW(I,1),D(I,1),I=1,N)                    CHE   40
      IF (ICHECK.EQ.1) RETURN KABORT                                    CHE   50
      WRITE (6,20) A0,F0,G0,A1,F1,G1                                    CHE   60
      IF (ICHECK.EQ.2) RETURN KABORT                                    CHE   70
      WRITE (6,30) U1,UU1,U2                                            CHE   80
      RETURN KABORT                                                     CHE   90
C                                                                       CHE  100
   10 FORMAT (20X,44HTHIS IS SUBROUTINE CHECK WHICH GIVES ALL THE,35H INCHE  110
     $FORMATION IN SUBROUTINE LINESCH.,//,30X,7HFNOW = ,1PE16.9,//,37X,4CHE  120
     $HXNOW,17X,1HD,//,=(25X,I2,3X,1PE16.9,4X,1PE16.9,/))               CHE  130
   20 FORMAT (//,20X,"A0 = ",1PE16.9,5X,"F0 = ",E16.9,5X,"G0 = ",E16.9,/CHE  140
     $20X,"A1 = ",E16.9,5X,"F1 = ",E16.9,5X,"G1 = ",E16.9)              CHE  150
   30 FORMAT (//,20X,5HU1 = ,1E16.9,5X,6HUU1 = ,1PE16.9,5X,5HU2 = ,1PE16CHE  160
     $.9)                                                               CHE  170
      END                                                               CHE  180
```

```
      SUBROUTINE CORNER (X,N,NDEC)                                      CCR   10
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25CCCR   20
     $,3),XC(15),XF(15),YC(15),YF(15),ZC(15),ZF(15),NPOSITS(15),RHO(573)CCR   30
     $,ISEG1,IPRCF,MAXIT,PENCPIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTRCPT,TMPLON(CCR   40
     $573),ORD(15),ITG,ICPT,IT,ZEPGLOB,NEWTRAJ,ARRAY(573,10)            CCR   50
      INTEGER ORD                                                       CCR   60
      LOGICAL ZERGLOB,NEWTRAJ                                           CCR   70
      DIMENSION X(NDEC,1)                                               CCR   80
      NVAR = 2                                                          CCR   90
      IF (IPRCF.EC.1) NVAR = 3                                          CCR  100
      DO 10 I = 1,NTRCPT                                                CCR  110
        NSEG1 = NSEG-1                                                  CCR  120
        DO 10 J = 1,NSEG1                                               CCR  130
        DO 10 K = 1,NVAR                                                CCR  140
        L = (I-1)*NVAR*NSEG1+(J-1)*NVAR+K                               CCR  150
        IF (K.EQ.1) XM(ORD(I),J+1) = X(L,1)                             CCR  160
        IF (K.EQ.2) YM(ORD(I),J+1) = X(L,1)                             CCR  170
        IF (K.EQ.3) ZM(ORD(I),J+1) = X(L,1)                             CCR  180
   10   CONTINUE                                                        CCR  190
      DO 30 I = 1,NTRAJ                                                 CCR  200
        WRITE (6,9010) I,XM(ORD(I),1),YM(ORD(I),1),ZM(ORD(I),1)         CCR  210
        DO 20 J = 2,NSEG                                                CCR  220
          WRITE (6,9020) XM(ORD(I),J),YM(ORD(I),J),ZM(ORD(I),J)         CCR  230
   20     CONTINUE                                                      CCR  240
        WRITE (6,9020) XM(ORD(I),NSEG+1),YM(ORD(I),NSEG+1),ZM(ORD(I),NSECCR  250
     $  G+1)                                                            CCR  260
```

```
   30  CONTINUE                                                          CCP  270
       RETURN                                                            CCP  280
C                                                                        CCP  290
 9010 FORMAT (12X,"FLIGHT PATH NO! ",I2,/,14X,"CORNER PTS.",/,14X,"(",F8CCP  300
     $.0,",",F8.0,",",F8.0,",",F8.0,")")                                 CCP  310
 9020 FORMAT (14X,"(",F8.0,",",F8.0,",",F8.0,",",F8.0,")")               CCP  320
       END                                                               CCP  330



       SUBROUTINE DYNAMID (PNALTY)                                       DYN   10
       COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,250DYN   20
      $,3),XO(15),XF(15),YC(15),YF(15),ZO(15),ZF(15),NPOSITS(15),PHO(573)DYN   30
      $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTROPT,TMPLON(DYN   40
      $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)           DYN   50
       INTEGER ORD                                                       DYN   60
       LOGICAL ZERGLOB,NEWTRAJ                                           DYN   70
       COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,ZWEIT1,GMAMIN(15),GMAMAX(15)DYN   80
      $,NCAMP                                                            DYN   90
       COMMON /CORNER/ ANGLE(15,12),POSIT(15,250,3),XCENTR(15,12),YCENTR(DYN  100
      $15,12),ZCENTR(15,12),XT1(15,12),YT1(15,12),ZT1(15,12),XT2(15,12),YDYN  110
      $T2(15,12),ZT2(15,12),NPOSIT(15),DIS(15)                           DYN  120
       REAL LENGTH                                                       DYN  130
       LENGTH(A,B,C,D,E,F) = SQRT((A-B)**2+(C-D)**2+(E-F)**2)            DYN  140
       PI = ATAN(1.)*4.                                                  DYN  150
C                                                                        DYN  160
C ..................................................................... DYN  170
C .                                                                   . DYN  180
C .   FIRST CALCULATE PARAMETERS AT CORNER POINTS                     . DYN  190
C .                                                                   . DYN  200
C ..................................................................... DYN  210
C                                                                        DYN  220
       MAX = NTROPT                                                      DYN  230
       IF (ZERGLOB) MAX = NTRAJ                                          DYN  240
C      INCLUDE ALL TRAJS FOR THE ZEROTH GLOBAL ITERATION                 DYN  250
       DO 60 II = 1,MAX                                                  DYN  260
         I = ORD(II)                                                     DYN  270
         DO 10 J = 2,NSEG                                                DYN  280
           YT2(I,J) = 0.                                                 DYN  290
           YT1(I,J) = YT2(I,J)                                           DYN  300
           XT2(I,J) = YT1(I,J)                                           DYN  310
           XT1(I,J) = XT2(I,J)                                           DYN  320
           ZT1(I,J) = 0.                                                 DYN  330
           ZT2(I,J) = C.                                                 DYN  340
           ANGLE(I,J-1) = XT1(I,J)                                       DYN  350
           YCENTR(I,J-1) = ANGLE(I,J-1)                                  DYN  360
           XCENTR(I,J-1) = YCENTR(I,J-1)                                 DYN  370
   10      CONTINUE                                                      DYN  380
         DIS(I) = 0.                                                     DYN  390
         XT1(I,1) = XM(I,1)                                              DYN  400
         XT2(I,1) = XM(I,1)                                              DYN  410
         YT1(I,1) = YM(I,1)                                              DYN  420
         YT2(I,1) = YM(I,1)                                              DYN  430
         ZT1(I,1) = ZM(I,1)                                              DYN  440
         ZT2(I,1) = ZM(I,1)                                              DYN  450
         ISEG2 = ISEG1+1                                                 DYN  460
         DO 40 J = ISEG2,NSEG                                            DYN  470
           A = LENGTH(XM(I,J-1),XM(I,J),YM(I,J-1),YM(I,J),0.,0.)         DYN  480
           B = LENGTH(XM(I,J),XM(I,J+1),YM(I,J),YM(I,J+1),C.,0.)         DYN  490
           C = LENGTH(XM(I,J-1),XM(I,J+1),YM(I,J-1),YM(I,J+1),0.,0.)     DYN  500
C                                                                        DYN  510
C ..................................................................... DYN  520
C .                                                                   . DYN  530
```

```
C .  ' DO THE TWO SEGMENTS FORM A STRAIGHT LINE?                          . DYN  540
C .                                                                      . DYN  550
C ..................................................................... DYN  560
C                                                                        DYN  570
      IF (ABS(A+B-C).GT.1.) GO TO 20                                     DYN  580
      SEG = LENGTH(XT2(I,J-1),XM(I,J),YT2(I,J-1),YM(I,J),ZT2(I,J-1),DYN  590
     $ ZM(I,J))                                                          DYN  600
      XT1(I,J) = XM(I,J)                                                 DYN  610
      XT2(I,J) = XM(I,J)                                                 DYN  620
      YT1(I,J) = YM(I,J)                                                 DYN  630
      YT2(I,J) = YM(I,J)                                                 DYN  640
      ZT1(I,J) = ZM(I,J)                                                 DYN  650
      ZT2(I,J) = ZM(I,J)                                                 DYN  660
      ZCENTR(I,J-1) = 0.                                                 DYN  670
      YCENTR(I,J-1) = 0.                                                 DYN  680
      XCENTR(I,J-1) = YCENTR(I,J-1)                                      DYN  690
      ALONG1 = 0.                                                        DYN  700
      ANGLE(I,J-1) = 0.                                                  DYN  710
      GO TO 30                                                           DYN  720
   20 AXY = LENGTH(XM(I,J-1),XM(I,J),YM(I,J-1),YM(I,J),0.,0.)            DYN  730
      BXY = LENGTH(XM(I,J),XM(I,J+1),YM(I,J),YM(I,J+1),0.,0.)            DYN  740
      CXY = LENGTH(XM(I,J-1),XM(I,J+1),YM(I,J-1),YM(I,J+1),0.,0.)        DYN  750
      BETA = AACOS(AXY,BXY,CXY)                                          DYN  760
      ALONG2 = RADIUS/TAN(BETA/2.)                                       DYN  770
      PNALTY = PNALTY+PWEIT1*((AMAX1(0.,(ALONG2-AXY)))**2+AMAX1(0.,(DYN  780
     $ ALONG2-BXY)))**2                                                  DYN  790
      IF (J.NE.ISEG2) PNALTY = PNALTY+PWEIT1*(AMAX1(0.,(ALONG1+ALONGDYN  800
     $ 2-AXY)))**2                                                       DYN  810
C                                                                        DYN  820
C ..................................................................... DYN  830
C .                                                                    . DYN  840
C .   CALCLLATE TANGENTIAL POINTS, CENTER OF RADIUS R AT J-TH CORNER   . DYN  850
C .                                                                    . DYN  860
C ..................................................................... DYN  870
C                                                                        DYN  880
      XT1(I,J) = XM(I,J)+ALONG2*(XM(I,J-1)-XM(I,J))/AXY                  DYN  890
      YT1(I,J) = YM(I,J)+ALONG2*(YM(I,J-1)-YM(I,J))/AXY                  DYN  900
      XT2(I,J) = XM(I,J)+ALONG2*(XM(I,J+1)-XM(I,J))/BXY                  DYN  910
      YT2(I,J) = YM(I,J)+ALONG2*(YM(I,J+1)-YM(I,J))/BXY                  DYN  920
      ZT1(I,J) = ZM(I,J)+ALONG2*(ZM(I,J-1)-ZM(I,J))/AXY                  DYN  930
      ZT2(I,J) = ZM(I,J)+ALONG2*(ZM(I,J+1)-ZM(I,J))/BXY                  DYN  940
      SEG = LENGTH(XT2(I,J-1),XT1(I,J),YT2(I,J-1),YT1(I,J),ZT2(I,J-1DYN  950
     $ ),ZT1(I,J))                                                       DYN  960
      DISM2 = ALONG2*COS(BETA/2.)                                        DYN  970
      DISC2 = RADIUS/SIN(BETA/2.)                                        DYN  980
      X = (XT1(I,J)+XT2(I,J))/2.                                         DYN  990
      Y = (YT1(I,J)+YT2(I,J))/2.                                         DYN 1000
      XCENTR(I,J-1) = XM(I,J)+DISC2*(X-XM(I,J))/DISM2                    DYN 1010
      YCENTR(I,J-1) = YM(I,J)+DISC2*(Y-YM(I,J))/DISM2                    DYN 1020
      ALONG1 = ALONG2                                                    DYN 1030
      D = LENGTH(XT1(I,J),XT2(I,J),YT1(I,J),YT2(I,J),0.,0.)              DYN 1040
      ANGLE(I,J-1) = AACOS(RADIUS,RADIUS,D)                              DYN 1050
      SEG = SEG+ANGLE(I,J-1)*RADIUS                                      DYN 1060
   30 DIS(I) = DIS(I)+SEG                                                DYN 1070
   40 CONTINUE                                                           DYN 1080
      DO 50 J = ISEG2,NSEG                                               DYN 1090
C                                                                        DYN 1100
C ..................................................................... DYN 1110
C .                                                                    . DYN 1120
C .   CALCLLATE VERTICAL SLOPE OF EACH SEGMENT AND                     . DYN 1130
C .   ADD PENALTY FOR EACH SEGMENT WITH VERTICAL SLOPE GREATER         . DYN 1140
C .   THAN GMAMAX OR LESS THAN GMAMIN DEGREES                          . DYN 1150
C .                                                                    . DYN 1160
C ..................................................................... DYN 1170
```

C-3

```
C     .                                                                 DYN 1180
          XYL = LENGTH(XM(I,J+1),XM(1,J),YM(I,J+1),YM(I,J),0.,0.)       DYN 1190
          VSLOPE = (ZM(I,J+1)-ZM(I,J))/XYL                              DYN 1200
          PNALTY = PNALTY+ZWEIT1*(AMAX1(0.,VSLOPE-TAND(GMAMAX(I))))**2+ZDYN 1210
         $    WEIT1*(AMAX1(0.,TAND(GMAMIN(I))-VSLOPE))**2              DYN 1220
       5C    CONTINUE                                                  DYN 1230
          AXY = LENGTH(XM(I,NSEG),XM(I,NSEG+1),YM(I,NSEG),YM(I,NSEG+1),0.,DYN 1240
         $  0.)                                                         DYN 1250
          BXY = LENGTH(XM(I,NSEG+1),XM(I,NSEG+2),YM(I,NSEG+1),YM(I,NSEG+2)DYN 1260
         $  ,0.,0.)                                                     DYN 1270
          CXY = LENGTH(XM(I,NSEG),XM(I,NSEG+2),YM(I,NSEG),YM(I,NSEG+2),0.,DYN 1280
         $  0.)                                                         DYN 1290
          ANG = AACOS(AXY,BXY,CXY)                                      DYN 1300
          PNALTY = PNALTY+PWEIT1*(AMAX1(0.,(BXY+ALONG1-AXY)))**2+PWEIT2*(ADYN 1310
         $  MAX1(0.,(2.3805-ANG)))**2                                   DYN 1320
          DIS(I) = DIS(I)+LENGTH(XT2(I,NSEG),XM(I,NSEG+1),YT2(I,NSEG),YM(IDYN 1330
         $  ,NSEG+1),ZT2(I,NSEG),ZM(I,NSEG+1))                         DYN 1340
       6C CONTINUE                                                      DYN 1350
          RETURN                                                        DYN 1360
          END                                                           DYN 1370


          SUBROUTINE NII (ANII)                                         NII   10
          COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25CNII   20
         $,3),XO(15),XF(15),YC(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)NII   30
         $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTRCPT,TMPLON(NII   40
         $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)        NII   50
          INTEGER ORD                                                   NII   60
          LOGICAL ZERGLOB,NEWTRAJ                                       NII   70
          PEOPLE = 0.                                                   NII   80
          ALWP = 0.                                                     NII   90
          DO 10 K = 1,NMAP                                              NII  100
            IF (ARRAY(K,3).LT.1.) GO TO 10                              NII  110
            ALON = TMPLON(K)                                            NII  120
            IF (ALON.LT.55.) GO TO 10                                   NII  130
            W = 3.36E-6*10.**(0.103*ALON)                               NII  140
            W = W/(0.2*10.**(0.03*ALON)+1.43E-4*10.**(0.08*ALON))       NII  150
            PEOPLE = PEOPLE+ARRAY(K,3)                                  NII  160
            ARRAY(K,5) = ARRAY(K,3)*W                                   NII  170
            ALWP = ALWP+ARRAY(K,5)                                      NII  180
       10   CONTINUE                                                    NII  190
          ANII = ALWP/PEOPLE*AWEIT                                      NII  200
          RETURN                                                        NII  210
          END                                                           NII  220


          SUBROUTINE NOISEL (THRESH)                                    NOI   10
          COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25CNOI   20
         $,3),XO(15),XF(15),YC(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)NOI   30
         $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTRCPT,TMPLON(NOI   40
         $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)        NOI   50
          INTEGER ORD                                                   NOI   60
          LOGICAL ZERGLOB,NEWTRAJ                                       NOI   70
          COMMON /CORNER/ ANGLE(15,12),POSIT(15,250,3),XCENTP(15,12),YCENTP(NOI   80
         $15,12),ZCENTP(15,12),XT1(15,12),YT1(15,12),ZT1(15,12),XT2(15,12),YNOI   90
         $T2(15,12),ZT2(15,12),NPOSIT(15),DIS(15)                       NOI  100
          COMMON /DYNAMIC/ RADIUS,PWEIT1,PWEIT2,ZWEIT1,GMAMIN(15),GMAMAX(15)NOI  110
         $,NSAMP                                                        NOI  120
          COMMON /THRESH/ ALMAX,MAXFLIT,TWEIT1,TWEIT2,NPLANE            NOI  130
          COMMON /SAMPLE/ SAMPT,PERIOD,SAMLTH,NAC(15,32),CNOIS(16,2)    NOI  140
```

```
      .DIMENSION ARCLTH(8), SEGLTH(9)                                          NOI  150
      REAL LENGTH                                                              NOI  160
      COORD(A,B,C,D) = (A/B)*(C-D)+D                                           NOI  170
      LENGTH(A,B,C,D,E,F) = SQRT((A-B)**2+(C-D)**2+(E-F)**2)                   NOI  180
      PI = ATAN(1.)*4.                                                         NOI  190
      MAX = 1                                                                  NOI  200
C                                                                              NOI  210
C ************************************************************************     NOI  220
C *                                                                    *     NOI  230
C *   INCLUDE ALL TRAJS FOR THE ZEROTH GLOBAL ITERATION                *     NOI  240
C *                                                                    *     NOI  250
C ************************************************************************     NOI  260
C                                                                              NOI  270
      IF (ZEROGLOB) MAX = NTRAJ                                                NOI  280
      DO 100 KK = 1,MAX                                                        NOI  290
        I = OPD(KK)                                                            NOI  300
        XT2(I,NSEG+1)=XM(I,NSEG+1)                                             NOI  310
        XT1(I,NSEG+1)=XT2(I,NSEG+1)                                            NOI  320
        YT2(I,NSEG+1)=YM(I,NSEG+1)                                             NOI  330
        YT1(I,NSEG+1)=YT2(I,NSEG+1)                                            NOI  340
        ZT2(I,NSEG+1)=ZM(I,NSEG+1)                                             NOI  350
        ZT1(I,NSEG+1)=ZT2(I,NSEG+1)                                            NOI  360
        DLZPM = (ZO(I)-ZF(I))/DIS(I)                                           NOI  370
        POSIT(I,1,1) = XM(I,ISEG1)                                             NOI  380
        POSIT(I,1,2) = YM(I,ISEG1)                                             NOI  390
        POSIT(I,1,3) = ZM(I,ISEG1)                                             NOI  400
        NSEG1 = NSEG-1                                                         NOI  410
        DO 10 J = ISEG1,NSEG1                                                  NOI  420
          SEGLTH(J) = LENGTH(XT2(I,J),XT1(I,J+1),YT2(I,J),YT1(I,J+1),ZT2NOI  430
     $    (I,J),ZT1(I,J+1))                                                    NOI  440
          ARCLTH(J) = RADIUS*ANGLE(I,J)                                        NOI  450
   10     CONTINUE                                                            NOI  460
        SEGLTH(NSEG) = LENGTH(XT2(I,NSEG),XM(I,NSEG+1),YT2(I,NSEG),YM(I,NOI  470
     $  NSEG+1),ZT2(I,NSEG),ZM(I,NSEG+1))                                     NOI  480
        II = 1                                                                 NOI  490
        DO 70 J = ISEG1,NSEG1                                                  NOI  500
   20     IF (SEGLTH(J).LT.SAMLTH) GO TO 30                                   NOI  510
          II = II+1                                                            NOI  520
          IF (II.GT.250) WRITE (6,9010) I,J,II                                 NOI  530
          IF (II.GT.250) STOP                                                  NOI  540
          POSIT(I,II,1) = COORD(SAMLTH,SEGLTH(J),XT1(I,J+1),POSIT(I,II-1NOI  550
     $    ,1))                                                                 NOI  560
          POSIT(I,II,2) = COORD(SAMLTH,SEGLTH(J),YT1(I,J+1),POSIT(I,II-1NOI  570
     $    ,2))                                                                 NOI  580
          POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                         NOI  590
          IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(SAMLTH,SEGLTH(J),ZT1(I,JNOI  600
     $    +1),POSIT(I,II-1,3))                                                 NOI  610
          SEGLTH(J) = SEGLTH(J)-SAMLTH                                         NOI  620
          GO TO 20                                                             NOI  630
   30     IF (SEGLTH(J)+ARCLTH(J).GE.SAMLTH) GO TO 40                         NOI  640
          PIECE = SAMLTH-ARCLTH(J)-SEGLTH(J)                                   NOI  650
          II = II+1                                                            NOI  660
          IF (II.GT.250) WRITE (6,9010) I,J,II                                 NOI  670
          IF (II.GT.250) STOP                                                  NOI  680
          POSIT(I,II,1) = COORD(PIECE,SEGLTH(J+1),XT1(I,J+2),XT2(I,J+1))NOI  690
          POSIT(I,II,2) = COORD(PIECE,SEGLTH(J+1),YT1(I,J+2),YT2(I,J+1))NOI  700
          POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                         NOI  710
          IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(PIECE,SEGLTH(J+1),ZT1(I,NOI  720
     $    J+2),ZT2(I,J+1))                                                     NOI  730
          SEGLTH(J+1) = SEGLTH(J+1)-PIECE                                      NOI  740
          GO TO 70                                                             NOI  750
   40     PIECE = SAMLTH-SEGLTH(J)                                            NOI  760
          GAMMA = PIECE/RADIUS                                                 NOI  770
          ARCLTH(J) = ARCLTH(J)-PIECE                                          NOI  780
```

```
      *     II = II+1                                                          NCI  790
            IF (II.GT.250) WRITE (6,9010) I,J,II                               NCI  800
            IF (II.GT.250) STOP                                                NCI  810
            ALFA1 = ATAN2((YT1(I,J+1)-YCENTR(I,J)),(XT1(I,J+1)-XCENTR(I,J))NCI  820
           $ ))                                                                NCI  830
            IPLUS = 1                                                          NCI  840
            AA = ALFA1+ANGLE(I,J)/4.                                           NCI  850
            X22 = XCENTR(I,J)+RADIUS*COS(AA)                                   NCI  860
            Y22 = YCENTR(I,J)+RADIUS*SIN(AA)                                   NCI  870
            ADIS = SQRT((X22-XM(I,J+1))**2+(Y22-YM(I,J+1))**2)                 NCI  880
            BDIS = SQRT((XT1(I,J+1)-XM(I,J+1))**2+(YT1(I,J+1)-YM(I,J+1))**NCI  890
           $ 2)                                                                NCI  900
            IF (BDIS.LT.ADIS) IPLUS = -1                                       NCI  910
            AA = ALFA1+IPLUS*GAMMA                                             NCI  920
            POSIT(I,II,1) = XCENTR(I,J)+RADIUS*COS(AA)                         NCI  930
            POSIT(I,II,2) = YCENTR(I,J)+RADIUS*SIN(AA)                         NCI  940
            POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI  950
            IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(PIECE,ARCLTH(J)+PIECE,ZTNCI  960
           $ 2(I,J+1),ZT1(I,J+1))                                              NCI  970
      50    IF (ARCLTH(J).LT.SAMLTH) GO TO 60                                  NCI  980
            AA = AA+IPLUS*SAMLTH/RADIUS                                        NCI  990
            ARCLTH(J) = ARCLTH(J)-SAMLTH                                       NCI 1000
            II = II+1                                                          NCI 1010
            IF (II.GT.250) WRITE (6,9010) I,J,II                               NCI 1020
            IF (II.GT.250) STOP                                                NCI 1030
            POSIT(I,II,1) = XCENTR(I,J)+RADIUS*COS(AA)                         NCI 1040
            POSIT(I,II,2) = YCENTR(I,J)+RADIUS*SIN(AA)                         NCI 1050
            POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI 1060
            IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(SAMLTH,ARCLTH(J)+SAMLTH,NCI 1070
           $ ZT2(I,J+1),POSIT(I,II-1,3))                                       NCI 1080
            GO TO 50                                                           NCI 1090
      60    PIECE = SAMLTH-ARCLTH(J)                                           NCI 1100
            II = II+1                                                          NCI 1110
            IF (II.GT.250) WRITE (6,9010) I,J,II                               NCI 1120
            IF (II.GT.250) STOP                                                NCI 1130
            POSIT(I,II,2) = COORD(PIECE,SEGLTH(J+1),YT1(I,J+2),YT2(I,J+1))NCI 1140
            POSIT(I,II,1) = COORD(PIECE,SEGLTH(J+1),XT1(I,J+2),XT2(I,J+1))NCI 1150
            POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI 1160
            IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(PIECE,SEGLTH(J+1),ZT1(I,NCI 1170
           $ J+2),ZT2(I,J+1))                                                  NCI 1180
            SEGLTH(J+1) = SEGLTH(J+1)-PIECE                                    NCI 1190
      70    CONTINUE                                                           NCI 1200
      80    IF (SEGLTH(NSEG).LT.SAMLTH) GO TO 90                               NCI 1210
            II = II+1                                                          NCI 1220
            IF (II.GT.250) WRITE (6,9010) I,J,II                               NCI 1230
            IF (II.GT.250) STOP                                                NCI 1240
            POSIT(I,II,1) = COORD(SAMLTH,SEGLTH(NSEG),XT1(I,NSEG+1),POSIT(I,NCI 1250
           $ II-1,1))                                                          NCI 1260
            POSIT(I,II,2) = COORD(SAMLTH,SEGLTH(NSEG),YT1(I,NSEG+1),POSIT(I,NCI 1270
           $ II-1,2))                                                          NCI 1280
            POSIT(I,II,3) = POSIT(I,II-1,3)-DLZPM*SAMLTH                       NCI 1290
            IF (IPROF.EQ.1) POSIT(I,II,3) = COORD(SAMLTH,SEGLTH(NSEG),ZT1(I,NCI 1300
           $ NSEG+1),POSIT(I,II-1,3))                                          NCI 1310
            SEGLTH(NSEG) = SEGLTH(NSEG)-SAMLTH                                 NCI 1320
            GO TO 80                                                           NCI 1330
      90    NPOSIT(I) = II                                                     NCI 1340
            IF (II.GT.250) CALL ERROR (6,'NO. OF TRAJ. SAMPLES .GT. 250,       NCI 1350
           $    RE-DIMENSION ARRAYS POSIT AND SPOSIT')                         NCI 1360
     100  CONTINUE                                                            NCI 1370
          NSAMP = IFIX(PERIOD*3600/SAMPT)                                      NCI 1380
          TVIOLA = 0.                                                          NCI 1390
          DO 110 K = 1,NMAP                                                    NCI 1400
            AMEN = 0.                                                          NCI 1410
C                                                                              NCI 1420
```

```
C ..................................................................... NCI 1430
C .                                                                   . NCI 1440
C .   CAL. ENERGY CONTRIB. BY AMBIENT NOISE,                         . NCI 1450
C .   CURRENTLY AMBIENT LEVEL = 0. DB                                 . NCI 1460
C .                                                                   . NCI 1470
C ..................................................................... NCI 1480
C                                                                       NCI 1490
      ARRAY(K,4) = NSAMP*10.**(AMBN/10.)                                NCI 1500
  110 CONTINUE                                                          NCI 1510
C                                                                       NCI 1520
C ..................................................................... NCI 1530
C .                                                                   . NCI 1540
C .   IF THIS IS THE ZEROTH GLOBAL ITERATION, INCLUDE                 . NCI 1550
C .   NOISE EFFECTS OF ALL TRAJS IN TMPLDN                            . NCI 1560
C .   ELSE, CALC. EFFECT OF TRAJ. BEING OPT. AND                      . NCI 1570
C .   STORE IN ARRAY(*,4)                                             . NCI 1580
C .                                                                   . NCI 1590
C ..................................................................... NCI 1600
C                                                                       NCI 1610
      IF (.NOT.ZERGLOB.OR.MAX.EQ.1) GO TO 160                           NCI 1620
      DO 150 K = 1,NMAP                                                 NCI 1630
        BLKVIOL = 0.                                                    NCI 1640
        IF (ARRAY(K,3).LT.1) GO TO 140                                  NCI 1650
        DO 130 II = 2,MAX                                               NCI 1660
          I = ORD(II)                                                   NCI 1670
          NPOSITI = NPOSIT(I)                                           NCI 1680
          ENERGY = 0.                                                   NCI 1690
          DO 120 J = 1,NPOSITI                                          NCI 1700
            RANGE = SQRT((POSIT(I,J,1)-ARRAY(K,1))**2+(POSIT(I,J,2)-ARRAYNCI 1710
     $      Y(K,2))**2+(POSIT(I,J,3))**2)                               NCI 1720
            AL = CNOIS(I,1)-CNOIS(I,2)*ALOG10(RANGE)                    NCI 1730
            ENERGY = ENERGY+10.**(AL/10.)                               NCI 1740
  120       CONTINUE                                                    NCI 1750
          TMPLDN(K) = ENERGY+TMPLDN(K)                                  NCI 1760
  130     CONTINUE                                                      NCI 1770
  140   TMPLDN(K) = 10.*ALOG10(TMPLDN(K)/(PERIOD*3600./SAMPT))         NCI 1780
        BLKVIOL = AMAX1(0.,BLKVIOL-ALMAX*MAXFLIT)                       NCI 1790
        IF (ARRAY(K,10).EQ.1.) TVIOLA = TVIOLA+BLKVIOL                  NCI 1800
  150   CONTINUE                                                        NCI 1810
  160 DO 180 K = 1,NMAP                                                 NCI 1820
        IF (ARRAY(K,3).LT.1) GO TO 180                                  NCI 1830
        I = ORD(1)                                                      NCI 1840
        NPOSITI = NPOSIT(I)                                             NCI 1850
        ENERGY = 0.                                                     NCI 1860
        DO 170 J = 1,NPOSITI                                            NCI 1870
          RANGE = SQRT((POSIT(I,J,1)-ARRAY(K,1))**2+(POSIT(I,J,2)-ARRAY(NCI 1880
     $    K,2))**2+(POSIT(I,J,3))**2)                                   NCI 1890
          AL = CNOIS(I,1)-CNOIS(I,2)*ALOG10(RANGE)                      NCI 1900
          ENERGY = ENERGY+10.**(AL/10.)                                 NCI 1910
  170     CONTINUE                                                      NCI 1920
        ARRAY(K,4) = 10.*ALOG10(ENERGY/(PERIOD*3600./SAMPT))            NCI 1930
  180   CONTINUE                                                        NCI 1940
C                                                                       NCI 1950
C ..................................................................... NCI 1960
C .                                                                   . NCI 1970
C .   ADD THE NOISE FROM TRAJ. BEING OPT. TO THE OTHER NOISE          . NCI 1980
C .                                                                   . NCI 1990
C ..................................................................... NCI 2000
C                                                                       NCI 2010
      IF (ZERGLOB.OR..NOT.NEWTRAJ) CALL ACOR (1)                        NCI 2020
      THRESH = TWEITI*TVIOLA**2                                         NCI 2030
      NEWTRAJ = .FALSE.                                                 NCI 2040
      RETURN                                                            NCI 2050
C                                                                       NCI 2060
```

```
 9010 FORMAT (*I = *,I2,* J = *,I2,* II = *,I10,///)                    NOI 2070
      END                                                               NOI 2080



      SUBROUTINE RCVRY (I,J,K)                                          RCV   10
      STOP                                                              RCV   20
      END                                                               RCV   30


      SUBROUTINE RNDROBN (MAXITG,STOPCHG,N,XNOW,DELTAX,NDEC)            RND   10
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25GRND   20
     $,3),XO(15),XF(15),YO(15),YF(15),ZO(15),ZF(15),NPOSITS(15),RHO(573)RND   30
     $,ISEG1,IPROF,MAXIT,PENCRIT,AWEIT,TVIOLA,ALWP,PEOPLE,NTROPT,TMPLON(RND   40
     $573),ORD(15),ITG,ICPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)            RND   50
      INTEGER ORD                                                       RND   60
      LOGICAL ZERGLOB,NEWTRAJ                                           RND   70
      DIMENSION XNOW(NDEC,1), DELTAX(NDEC)                              RND   80
      NSEG1 = NSEG-1                                                    RND   90
      DO 30 KK = 1,MAXITG                                               RND  100
        ITG = KK                                                        RND  110
        DO 30 I = 1,NTROPT                                              RND  120
        IOPT = I                                                        RND  130
        NEWTRAJ = .TRUE.                                                RND  140
        WRITE (6,9010) (ORD(J),J=1,NTRAJ)                               RND  150
        CALL ONEWTON (N,XNOW,DELTAX,STOPCHG,NDEC)                       RND  160
C                                                                       RND  170
C .....................................................................RND  180
C .                                                                   .RND  190
C .   TMPLON NOW CONTAINS NOISE EFFECTS FROM THE CONSTANT TRAJS,      .RND  200
C .   SO ADD IN THE NOISE FROM THE TRAJ. BEING OPT.                   .RND  210
C .                                                                   .RND  220
C .....................................................................RND  230
C                                                                       RND  240
        CALL ADDR (1)                                                   RND  250
        IF (NTROPT.EQ.1) RETURN                                         RND  260
C                                                                       RND  270
C .....................................................................RND  280
C .                                                                   .RND  290
C .   CYCLE THE ORDER OF THE TRAJS                                    .RND  300
C .                                                                   .RND  310
C .....................................................................RND  320
C                                                                       RND  330
        N1 = NTROPT-1                                                   RND  340
        ORDTMP = ORD(1)                                                 RND  350
        DO 10 K = 1,N1                                                  RND  360
          ORD(K) = ORD(K+1)                                             RND  370
   10     CONTINUE                                                      RND  380
        ORD(NTROPT) = ORDTMP                                            RND  390
        DO 20 J = 1,NSEG1                                               RND  400
          DO 20 K = 1,3                                                 RND  410
          L = (J-1)*3+K                                                 RND  420
          IF (K.EQ.1) XNOW(L,1) = XM(ORD(1),J+1)                        RND  430
          IF (K.EQ.2) XNOW(L,1) = YM(ORD(1),J+1)                        RND  440
          IF (K.EQ.3) XNOW(L,1) = ZM(ORD(1),J+1)                        RND  450
   20     CONTINUE                                                      RND  460
   30   CONTINUE                                                        RND  470
      WRITE (6,9020) NTROPT,MAXITG                                      RND  480
      RETURN                                                            RND  490
C                                                                       RND  500
 9010 FORMAT (/,5X,"ORDER OF TRAJS: ",15(I2,2X))                        RND  510
```

```
 9020 FORMAT (//,5X,"ROUND ROBIN PROCESS COMPLETE.  EACH OF THE ",I2," TRPND  520
     $AJS OPTIMIZED ",I2," TIME(S).")                                      RND  530
      END                                                                  RND  540



      SUBROUTINE ADDR (ISIGN)                                              ADD   10
      COMMON NTRAJ,NMAP,NSEG,XM(15,13),YM(15,13),ZM(15,13),SPOSIT(15,25ØADD   20
     $,3),XO(15),XP(15),YO(15),YF(15),ZC(15),ZF(15),NPOSITS(15),PHO(573)ADD   30
     $,ISEG1,IPROF,MAXIT,PERMIT,AWEIT,TVIOLA,ALWP,PECPLE,NTROPT,TMPLON(ADD   40
     $573),OFO(15),ITG,IGPT,IT,ZERGLOB,NEWTRAJ,ARRAY(573,10)              ADD   50
      INTEGER ORD                                                          ADD   60
      LOGICAL ZERGLOB,NEWTRAJ                                              ADD   70
      CALL SECOND (TO)                                                     ADD   80
      DO 10 K = 1,NMAP                                                     ADD   90
        E1 = 10.**(ARRAY(K,4)/10.)                                         ADD  100
        E2 = 10.**(TMPLON(K)/10.)                                          ADD  110
        ENEW = E2+ISIGN*E1                                                 ADD  120
        TMPLON(K) = 10.*ALOG10(ENEW)                                       ADD  130
   10   CONTINUE                                                          ADD  140
      RETURN                                                               ADD  150
      END                                                                  ADD  160
```

REFERENCES

1. Pearsons, K. S. and Bennett, R. L., Handbook of Noise Ratings, NASA CR-2376 (1974).

2. Alexandre, A., "An Assessment of Certain Causal Models Used in Surveys of Aircraft Noise Annoyance," Journal of Sound Vibration, Vol. 44, 1 (January 1976), pp. 119-125.

3. Tarnopolsky, A., "Effects of Aircraft Noise on Mental Health," Journal of Sound and Vibration, Vol. 59, 1 (July 1978), pp. 89-97.

4. Borsky, P. N., "A New Field-Laboratory Methodology for Assessing Response to Noise," NASA CR-2221 (March 1973).

5. Kryter, K. D., "The Effects of Noise on Man," Academic Press, Inc., New York, N. Y. (1970).

6. von Gierke, H. E., "Guidelines for Environmental Impact Statements with Respect to Noise," NOISE-CON 77 Proceedings, Noise Control Foundation, New York, N. Y. (1977), pp. 61-78.

7. Goldstein, J., "Assessing the Impact of Transportation Noise: Human Response Measures," NOISE-CON 77 Proceedings, Noise Control Foundation, New York, N. Y. (1977), pp. 79-98.

8. Schultz, T. J., "Synthesis of Social Surveys on Noise Annoyance," Journal of the Acouistical Society of America, Vol. 64, 2 (August 1978), pp. 377-405.

9. "Noise Source Abatement Technology and Cost Analysis Including Retrofitting," Environmental Protecton Agency Aircraft/Airport Noise Study Report, NTID 73.5 (July 27, 1973).

10. Padula, S. L. and Liu, C. H., "Acoustic Scattering of Point Sources by a Moving Prolate Spheroid," AIAA Journal, Vol. 16, 6 (June 1978), pp. 551-552.

11. Barkhana, A. and Cook, G., "An Aircraft Noise Pollution Model for Trajectory Optimization," IEEE Transactions on Aerospace and Electonic Systems, AES-12, 3 (March 1976), pp. 109-115.

12. Dunn, D. G. and Peart, N. A., "Aircraft Noise Source and Contour Estimation," NASA CR-114649 (July 1973).

13. Crowley, K. C., Jaeger, M. A., and Meldrum, D. F., "Aircraft Noise Source and Contour Computer Programs User's Guide," NASA CR-114650 (July 1973).

14. Filotas, L. T., "Effects of Flight Path Dispersion on Airport Noise," Journal of Sound and Vibration, Vol. 48, 4 (October 22, 1976), pp. 451-460.

15. Zalovcik, J. A., "Effect of Thrust and Altitude in Steep Approaches on Ground Track Noise," NASA TN D-4241 (November 1967).

16. Glick, J. M., Shevell, R. S., and Bowles, J. V., "Evaluation of Methods of Reducing Community Noise Impact Around San Jose Minicpal Airport," NASA TM X-62,503 (November 1975).

17. Jacob, H. G., "An Engineering Optimization Method With Application to STOL-Aircraft Approach and Landing Trajectories," NASA TN D-6978 (September 1972).

18. Barkhana, A., Cook, G., and Witt, R., "Aircraft Landing Trajectories of Minimum Noise," Mundo Electronico, 91 (December 1979), pp. 81-86.

19. Rader, J. E. and Hull, D. G., "Computation of Optimal Aircraft Trajectories Using Parameter Optimization Methods," Journal of Aircraft (November 1975).

20. Jacobson, I. D., Cook, G., Chang, R., and Melton, R., "Methodology for Multi-Aircraft Minimum Noise Impact Landing Trajectories," accepted for publication by the IEEE Transactions of Aerospace and Electronic Systems.

21. Jacobson, I. D. and Cook, G., "Evaluating the Minimizing Noise Impact Due to Aircraft Flyover," RLES Report No. UVA/528166/MAE81/102, University of Virginia (May 1980).

22. Erzberger, H., Lee, Homer, "Constrained Optimum Trajectories with Specified Range," Journal of Guidance and Control, Vol. 3, 1 (January-February 1980), pp. 78-85.

23. Luenberger, D. G., Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Co., Reading, MA (1973).

24. Integrated Noise Model, Data Base, Federal Aviation Administration, Version 2, released July 31, 1979.

25. Anonymous, SITE II User's Manual, CACI, Inc., Arlington, VA (1976).

26. Chang, R., Optimal Aircraft Landing Patterns for Minimal Noise Impact, Ph.D. Dissertation, University of Virginia (1981).

27. "Guidelines for Preparing Evironmental Impact Statements on Noise," Report of Working Group 69 on Evaluation of Environmental Impact of Noise, National Academy of Sciences, Washington, DC (1977).

28. Jacobson, I. D., "Constraint Descriptions," private notes, Department of Mechanical and Aerospace Engineering, University of Virginia (1979).

29. Jacobson, I.D., "Dynamic Stability of Aerospace Vehicles," lecture notes, Department of Mechanical and Aerospace Engineering, University of Virginia (1979).

30. Jacobson, I. D., Kulthau, A. R., Richards, L. G., and Conner, D. W., "Passenger Ride Quality in Transportation Aircraft," Journal of Aircraft, Vol. 15, 11, (1978) pp. 724-730.

31. Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or Least Value of a Function," The Computer Journal, 3, (1960), pp. 175-184.

32. Pierre, D. A., Optimization Theory with Applications, John Wiley and Sons, Inc. (1969).

33. Davidon, W. C., "Variable Metric Method for Minimization," Research and Development Report ANL-5990 (Rev.), U.S. Atomic Energy Commission, Argonne National Laboratories (1959).

34. Fletcher, R. and Powell, M. J. D., "A rapidly convergent descent method for minimization," The Computer Journal 6 (1963), pp. 163-168.

35. Bocharov, I. N. and Fel'dbaum, A. A., "An Automatic Optimizer for the Search of the Smallest of Several Minima (a Global Optimizer)," Automation and Remote Control, Vol. 23, 3 (1962), pp. 260-270.

36. Törn, Aimo A., "A Search-Clustering Approach to Global Optimization," Towards Global Optimization 2, paper no. 4, L. C. W. Dixon and G. P. Szegö (eds.), North-Holland Publishing Company (1978).

37. Galloway, W. J., "Community Noise Exposure Resulting from Aircraft Operations: Technical Review," Aerospace Medical Research Laboratory, AMRL-TR-73-106 (1974).

38. Deloach, R., Langley Research Center, Hampton, VA, private communication.

39. Official Airline Guide, North American Edition, Vol. 5, 1, R. H. Donnelly Publications (October 1, 1978).