NASA Contractor Report 159070

NASA-CR-159070

1979 0018640

NEWSUMT - A Fortran Program for Inequality Constrained Function Minimization - Users Guide

Hirokazu Miura and Lucien A. Schmit, Jr.

University of California, Los Angeles
Los Angeles, California 90024

Grant NGR-05-007-337
June 1979

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

Table of Contents

N79-26811 #

LIST OF FIGURES

SUMMARY

NEWSUMT is a computer program written in FORTRAN subroutine form for the solution of linear and nonlinear constrained or unconstrained function minimization problems. The basic algorithm is the sequence of unconstrained minimizations (Ref. 1) using the modified Newton's method (Ref. 2) for unconstrained function minimizations.

Problems must be formulated in the following form:

Minimize $F(X_1, X_2, \ldots X_n)$

Subject to $g_q(X_1, X_2, \ldots X_n) \geq 0 \quad q = 1,2,\ldots Q$

The user must provide a main program which calls subroutine NEWSUM and also subroutine ANALYS which computes the function values $F(\vec{X})$ and $g_q(\vec{X})$. If analytic gradients of these functions are available, the gradients should also be computed by ANALYS; otherwise the gradients will be computed by finite differences. Even if constraint functions or the objective function are not defined for certain values of the design variables, artificial definitions must be specified so that all functions are defined and differentiable over the entire design space.

This report describes the use of NEWSUMT and defines all necessary parameters. Sufficient information is provided so that the program can be used without special knowledge of nonlinear mathematical programming methods.

I. INTRODUCTION

NEWSUMT is a computer program written in a FORTRAN subroutine form for the solution of linear and nonlinear inequality constrained or unconstrained function minimization problems. The purpose of NEWSUMT is to determine the values of a set of variables $\vec{X}$ (a vector of real variables, $X_1, X_2, \ldots X_{NDV}$) that minimize a function $F(\vec{X})$ subject to a set of inequality constraints $g_q(\vec{X}) \geq 0$, $q = 1,2,\ldots,Q$. NEWSUMT was originally developed as an optimizer for sizing minimum weight finite element structural systems in the ACCESS-2 computer program (Ref. 3). However, it is a general purpose optimizer that can be used for solving a wide variety of numerical optimization problems. It treats inequality constraints in a way that is especially well suited to engineering design applications. In a structural design application, for example, $\vec{X}$ could represent dimensions of structural members, $F(\vec{X})$ could be the weight of the structure, and the $g_q(\vec{X})$, $q = 1,2,\ldots Q$, could express stress, buckling, or other types of behavioral constraints.

When using NEWSUMT the optimization problem must be formulated in the following form:

Minimize the objective function $F(\vec{X})$

Subject to inequality constraints

$$g_q(\vec{X}) \geqq 0, \quad q = 1,2,\ldots Q \tag{1}$$

$$X_j^{(L)} \leqq X_j \leqq X_j^{(U)}, \quad j = 1,2,\ldots NDV$$

where the functions $F(\vec{X})$ and $g_q(\vec{X})$ are continuous and differentiable real functions with respect to the design variables X_j, $j = 1,2,\ldots,NDV$. The user must supply a main program which calls NEWSUMT and also a subroutine which is called by the NEWSUMT program to evaluate functions $F(\vec{X})$, $g_q(\vec{X})$ and, if available, the derivatives of F and/or g_q with respect to the variables, X_j. The user specifies an initial design by assigning certain numerical values to $(X_1,X_2,\ldots,X_{NDV})$; the NEWSUMT program then systematically modifies these values generating a sequence of vectors $\vec{X}$ such that $F(\vec{X})$ decreases and none of the inequality constraints are critical. This sequence of vectors $\vec{X}$ converges to a solution $\vec{X}^*$ where all the inequality constraints are satisfied and $F(\vec{X}^*)$ is at least a local minimum.

II. MINIMIZATION ALGORITHM

In this section, the algorithms used in the NEWSUMT code are explained. All of these computations are performed internally. They are described here for completeness.

The minimization algorithm used in NEWSUMT is a sequence of unconstrained minimizations technique (SUMT), Ref. 1. Major features of NEWSUMT which distinguish it from the original formulation (see Ref. 1) include:

(a) A modified Newton's method is used in the direction finding part of the unconstrained minimization. In this method, second derivatives of the constraints are approximated by expressions involving only the first derivatives (Ref. 2)

(b) An extended interior penalty function formulation. This type of penalty function combines the features of interior and exterior penalty functions. That is, although initial designs may violate the constraints, subsequent designs satisfy the constraints and tend to be noncritical (Refs. 4,5). The transition point control parameter, which is a critical factor for numerical stability when using extended penalty functions, is selected so that the one dimensional search problems generated usually have their minimum points inside the feasible region (Ref. 6).

Since the detailed mathematical aspects of the SUMT algorithm and its variations are beyond the scope of this report, only matters of critical importance to understanding the fundamental procedure used in the NEWSUMT program will be described in the following subsections.

2.1 SUMT Approach

The SUMT algorithm transforms the inequality constrained problem defined by equations (1) into a sequence of unconstrained problems. To accomplish this transformation, a compound function $\phi(\vec{X},r_p)$ is introduced. The compound function used in NEWSUMT is defined as

$$\phi(\vec{X},r_p) = F(\vec{X}) + r_p \left[\sum_{q=1}^{Q} \frac{1}{g_q(\vec{X})} + \sum_{j=1}^{NDV} \left(\frac{1}{X_j - X_j^{(L)}} + \frac{1}{X_j^{(U)} - X_j} \right) \right] \tag{2}$$

In the transformed problem, $\phi(\vec{X},r_p)$ is minimized with respect to $\vec{X}$ for a sequence of decreasing values of r_p, which is called the penalty multiplier. Because r_p is being decreased, the contribution of the penalty function is being reduced and the solution to the transformed problem is converging toward the solution of the initial problem defined by Equation (1). Fundamental theory, convergence characteristics and various methods for solving SUMT-type problems are discussed in Ref. 1.

Optimization technology developed during recent years has greatly improved the computational efficiency and numerical stability of SUMT-type formulations. NEWSUMT incorporates many of these new features and they enhance its performance.

2.2 Extended Penalty Function

It must be noted that the composite function given by Eq. (2) is only defined for that portion of the design space where all the inequality constraints are satisfied. This is extremely inconvenient because it requires that an initial design which satisfies all constraints with a reasonable margin be available. Furthermore, since $\phi(\vec{X},r_p)$ is undefined in the infeasible region, the one dimensional minimization algorithm becomes complicated and inefficient. Using the extended penalty function concept, Eq. (2) is modified as follows

$$\phi(\vec{X},r_p) = F(\vec{X})+r_p\left[\sum_{q=1}^{Q} H_q(\vec{X}) + \sum_{j=1}^{NDV} [L(X_j) + U(X_j)]\right] \tag{3}$$

where

$$\vec{H}_q(X) = \begin{cases} \dfrac{1}{g_q(\vec{X})} & g_q(\vec{X}) \geq \varepsilon \\[2ex] \dfrac{1}{\varepsilon}\left[\left(\dfrac{g_q(\vec{X})}{\varepsilon}\right)^2 - \dfrac{3g_q(\vec{X})}{\varepsilon} + 3\right] & g_q(\vec{X}) < \varepsilon \end{cases} \tag{4}$$

$$U(X_j) = \begin{cases} 0 & X_j^{(L)} = - \text{ (unbounded)} \\[2ex] \dfrac{1}{X_j - X_j^{(L)}} & X_j = X_j^{(L)} \\[2ex] \dfrac{1}{\varepsilon}\left[\left(\dfrac{X_j-X_j^{(L)}}{\varepsilon}\right)^2 - \dfrac{3(X_j-X_j^{(L)})}{\varepsilon} + 3\right] & X_j < X_j^{(L)} \end{cases} \tag{5}$$

$$
U(X_j) = \begin{cases} 0 & X_j^{(U)} = +\infty \text{(unbounded)} \\ \dfrac{1}{X_j^{(U)} - X_j} & X_j = X_j^{(U)} \\ \dfrac{1}{\varepsilon}\left[\left(\dfrac{X_j^{(U)} - X_j}{\varepsilon}\right)^2 - \dfrac{3(X_j^{(U)} - X_j)}{\varepsilon} + 3\right] & X_j > X_j^{(U)} \end{cases} \tag{6}
$$

and ε denotes the transition parameter (Sec. 2.3). The advantage of the extended penalty function concept becomes apparent upon examining Fig. 3. The original penalty function defined by Eq. (2) is shown as a broken line which approaches infinity as $g_q(\vec{X})$ approaches zero from the positive side and it is not defined in the infeasible region where $g_q(\vec{X}) < 0$. On the other hand, the penalty function $H_q(\vec{X})$ given by Eq. (4) is defined for negative values of $g_q(\vec{X})$ and it is a smooth function (continuous up to the second derivatives) of $g_q(\vec{X})$. As a consequence, $H_q(\vec{X})$ is a well behaved smooth function within the region where $g_q(\vec{X})$ is defined and it is also a smooth function of $(\vec{X})$. An interesting interpretation of quadratic extended penalty function is given in Appendix A, where it is viewed as a combined interior-exterior penalty function approach.

2.3 Transition Parameter

The transition parameter (ε), introduced when defining extended penalty functions (Eqs. 4 through 6), is selected initially by the user (default value is 0.1) and may be given as input data. As soon as a design which satisfies all constraints is found after two or more unconstrained minimizations, the value assigned to ε is automatically estimated by the method set forth in Ref. 6. This method guarantees that the transition parameter is chosen as large as possible (to maintain numerical stability) while at the same time ensuring the minimum remains inside the feasible region. Once ε is determined automatically, the coefficient C which relates ε with r_p

$$\varepsilon = C\sqrt{r_p} \qquad (7)$$

is computed; then Eq. (7) is used thereafter to compute ε from the response factor r_p.

2.4 Modified Newton's Method

Minimization of $\phi(\vec{X}, r_p)$ with respect to $\vec{X}$ involves repeated application of two basic steps;

Step 1. Find a direction vector $\vec{S}$ along which the design is modified starting from the current design $\vec{X}_0$. A new design $\vec{X}$ is given by

$$\vec{X} = \vec{X}_0 + \alpha\vec{S} \qquad (8)$$

where α is a scalar variable that governs the move distance in the design space.

Step 2. Find the value for α so that the composite function $\phi(\vec{X}, r_p)$ is minimized along the direction $\vec{S}$.

The modified Newton's method discussed in this subsection deals with the procedure used to find a direction vector $\vec{S}$ (Step 1). In Newton's method, the direction vector $\vec{S}$ is given by

$$\vec{S} = - [J]^{-1} \nabla\phi / ||[J]^{-1}\nabla\phi|| \tag{9}$$

where [J] is a NDV x NDV matrix with (i,j) element defined by

$$J_{ij} = \frac{\partial^2 \phi}{\partial X_i \partial X_j} (\vec{X}, r_p) \tag{10}$$

Equation (10) may be evaluated by differentiating Eq. (3).

$$\frac{\partial^2 H_q}{\partial X_i \partial X_j} (\vec{X}) = \begin{cases} 2\left[\frac{\partial g_q}{\partial X_i} \frac{\partial g_q}{\partial X_j} - g_q \frac{\partial^2 g_q}{\partial X_i \partial X_j}\right] / g_q^{\,3} & g_q(\vec{X}) \geq \varepsilon \\ 2\left[\frac{\partial g_q}{\partial X_i} \frac{\partial g_q}{\partial X_j} + (2g_q - 3\varepsilon) \frac{\partial^2 g_q}{\partial X_i \partial X_j}\right] / \varepsilon^3 & g_q(\vec{X}) < \varepsilon \end{cases} \tag{11}$$

Following the suggestion in Ref. 2, $H_q(\vec{X})$ is simplified by neglecting the terms involving $\partial^2 g_q(X)/\partial X_i \partial X_j$, hence

$$\frac{\partial^2 H_q}{\partial X_i \partial X_j}(X) \cong \begin{cases} 2 \frac{\partial g_q}{\partial X_i} \frac{\partial g_q}{\partial X_j} / g_q^{\,3} & g_q(\vec{X}) \geq \varepsilon \\ 2 \frac{\partial g_q}{\partial X_i} \frac{\partial g_q}{\partial X_j} / \varepsilon^3 & g_q(\vec{X}) < \varepsilon \end{cases} \tag{12}$$

This approximation is justified qualitatively in the following manner. For critical constraints, $g_q(\vec{X})$ is small, therefore

$$2 \frac{\partial g_q}{\partial X_i} \frac{\partial g_q}{\partial X_j} >> g_q \frac{\partial^2 g_q}{\partial X_i \partial X_j} \tag{13}$$

assuming that $g_q(\vec{X})$ is a smooth function of $\vec{X}$. For noncritical constraints, $g_q(\vec{X})$ is large, thus the entire term corresponding to such constraint is small (due to $g_q^{\,3}$ in the denominator) compared to those associated with critical or nearly critical constraints.

The modified Newton's method uses these approximate contributions to the Hessian matrix (see Eq. 10) in computing the direction vector $\vec{S}$. For optimization problems involving a large number of complicated nonlinear constraints, experience confirms that this approach is efficient and generates good quality direction vectors so that only 4 - 6 one dimensional minimizations are sufficient for each unconstrained minimization, regardless of the number of design variables. If it is observed that the direction vector $\vec{S}$ found by the modified Newton's method does not decrease the composite function ϕ, $\vec{S}$ is replaced by the direction of the steepest descend, i.e. $-\nabla\phi/\|\nabla\phi\|$. This is experienced occasionally due to numerical ill-conditioning of the approximated Hessian matrix.

2.5 One Dimensional Minimization

As mentioned in the previous subsection, it is necessary to find the value of α for Eq. (8) such that $\phi(\vec{X},r_p)$ is minimized along the direction $\vec{S}$. This is achieved by first trapping a minimum in a finite interval and subsequently by applying the golden section algorithm to determine α_{min} with sufficient precision. First the move distance implied by the modified Newton method is determined. Then the distances to the hyperplanes representing linear constraints are calculated. The first trial stepsize is then taken as the smallest of the foregoing move distances. If the first trial design gives a smaller value for ϕ than the initial design, then the step size is increased by 2.6180 but it is not allowed to exceed the distance to the nearest linear constraint hyperplane. This process is continued until ϕ becomes greater than the previous evaluation, thus a minimum is trapped within a finite α interval. For the example shown in Fig. 5, a minimum is trapped after the third trial design is evaluated. Then the golden section algorithm is activated to find a minimum between α_2 and α_3.

2.6 Convergence Criteria

The golden section algorithm used to calculate α in one-dimensional minimizations is terminated if the maximum relative difference among the four function values used in the current step of golden section is smaller than the specified value (EPSGSN).

An unconstrained minimization is judged to be converged if two successive one-dimensional minimizations do not improve the compound function more than the specified fraction (EPSODM). Then, r_p is reduced by $r_{p+1}=r_p \times$ RPCUT, and another unconstrained minimization is initialized. The entire process is judged to be converged if two successive unconstrained minimizations do not improve the objective function more than the specified fraction (EPSRSF).

III. MAKING NEWSUMT OPERATIONAL

In order to become familiar with the NEWSUMT optimization code, the following steps are recommended.

1) Obtain the source program deck and sample problem decks.
2) Read Sec. IV. of this manual for example problems.
3) Execute the two sample problems. Both problems are self-contained and need no input data.
4) Read this entire manual.
5) Devise several two to five variables unconstrained and constrained minimization problems and solve them using NEWSUMT. If the correct optima can be determined analytically compare these with optima obtained using NEWSUMT.
6) Experiment with the various options available.

IV. PROGRAMMER'S GUIDE

4.1 Main Program

The overall organization is shown schematically in Fig. 1. Usually, all data must be read in the main program (or at least before NEWSUM is called). The primary subroutine of the NEWSUMT optimizer is activated by:

```
CALL NEWSUM*(name
*   ,BL    ,BU      ,DDOBJ  ,DG       ,DH  ,DOBJ
*   ,FDCV  ,FMIN    ,G      ,GB       ,G1  ,G2    ,G3
*   ,OBJ   ,OBJMIN  ,S      ,SN       ,X         ,XØ
*   ,IIK   ,ILIN    ,ISIDE  ,NI       ,N2  ,N3    ,N4
*   ,RAN   ,NRANDM  ,IAN    ,NIANDM)
```

where

name	:	Name of main subroutine of the analysis program which is called by NEWSUMT program. 6 characteris or less.
BL(N1)	:	Lower bounds imposed on design variables.
BU(N1)	:	Upper bounds imposed on design variables.
DDOBJ(N3)	:	Second order derivative matrix of the objective function. Only upper triangle is stored in a vector form; from top to the diagonal for each column.
DG(N4)	:	Gradient of constraints. Note that the storage scheme is:

$$g_{1,1}g_{2,1},\ldots g_{NTCE,1}g_{1,2}\ldots g_{NTCE,NDV}$$

*Variables with single underline should be assigned or initialized in the main program. Variables with double underlines are evaluated in the analysis program.

where

$$g_{q,i} = \frac{\partial g_q}{\partial X_i}$$

DH(N1) : Internally necessary array (gradient of the compound function).

DOBJ(N1) : Gradient of the objective function

FDCV(N1) : Stepsize of the forward finite difference steps for each design variable.

FMIN : Minimum of the composite function.

G(N2) : Constraint values.

GB(n2)
G1(N2)
G2(N2)
G3(N2) : Internally necessary array to store constraint function values.

S (N1) : Internally necessary array to store the direction vector.

X(N1) : Internally necessary array to store alternative design.

XØ(N1) : Initial design variables

IIK(N1) : IIK(i)=i(i+1)/2

ILIN(N2) : Linear constraint indicator
=1 for linear constraints
=0 for nonlinear constraints

ISIDE(N1) : Side constraint identification code
=+1 for lower bound only
= 0 no side constraint
=+2 for upper bound only
=+3 for lower and upper bounds.

N1 : =NDV Number of design variables

N2 : =NCON Number of constraints (=1, if NCON=0)

N3 : NDV(NDV+1)/2

N4 : NDV*NCON (=1, if NCON=0)

RAN(NRANDM) : Real array which may be used in the analysis program written by the user.

NRANDM : Dimension of RAN

IAN(NIANDM) : Integer array which may be used in the analysis program written by the user.

NIANDM : Dimension of IAN.

Note that all the variables or arrays with single underline must have their values assigned in the main program before the NEWSUM subroutine is called. Arrays with the double underlines receive specific values in the analysis program written by the user.

In addition to the variables transferred through the argument list of the NEWSUM subroutine, a labeled common block is used to transfer optimizer control parameters. The common block /CONTRL/ must be declared in the main program as

```
      COMMON/CONTRL/C, EPSGSN,EPSODM,EPSRSF,GØ     ,P
     * ,RA   ,RACUT , RAMIN ,STEPMX,
       ,IFD  ,JRPINT, JSIGNG,LOBJ  ,MAXGSN,MAXODM,MAXRSF
       ,MFLAG,NDV   , NTCE
```

where C : Not required to assign values.

EPSGSN : Convergence criteria of the golden section algorithm. For the golden section algorithm, there are always four function values to be compared to each other. If the sum of relative difference of these four values is smaller than EPSGSN, then the golden section is terminated. Default=0.001.

EPSODM : Convergence criteria of the unconstrained minimization. If 3 successive one-dimensional minimizations do not achieve relative improvements of the composite function or more than EPSODM, then unconstrained minimization is judged to have converged. Default=0.001.

EPSRSF : Convergence criteria for the overall process. If three successive unconstrained minimizations do not achieve relative improvements of the objective function of more than EPSRSF, then the NEWSUMT terminates the search process and returns control to the main program. Default = 0.005.

GØ : Initial value of the transition parameter. Default=0.01

P : =0.5. Default=0.5. Do not change

RA : Penalty multiplier r_p. Required if MFLAG=1. Default=1.0.

RACUT : Penalty multiplier decrease ratio. $r_{p+1}=r_p \times$ RACUT. Default=0.1

RAMIN : Lower bound of the penalty multiplier. If this is zero, numerical instability or excessive number of iterations may take place. Default=10^{-13}.

STEPMX : Maximum bound imposed on the initial step size of the one-dimensional minimization. Default=2.0. Note that the direction vector $\vec{S}$ is normalized so that $\vec{S}^T \cdot \vec{S}=1.0$ prior to each one-dimensional minimization.

IFD : Flag for finite difference gradient control.

= 0 All gradient information must be computed by the user's analysis program. The analysis program should accept INFO=3,4, and 5. Default.

> 0 Use default finite difference stepsize (0.01)

< 0 Use user supplied finite difference stepsize.

FDCV(i), i=1, NDV must be specified in the main program.

=1 Gradient of objective function must be computed by finite difference.

=2 Gradient of all constraints (including linear constraints*) must be obtained by finite difference.

=3 Gradient of nonlinear constraints must be by finite difference.

=4 1 and 2 combined.

=5 1 and 3 combined

JPRINT : Printout control parameter.

=0 Print initial and final designs only

=1 Print brief results of analysis for initial and final designs together with minimal intermediate information. (the default option is 1).

=2 Detailed printing.

=3 Debugging printing.

JSIGNG : Not used.

LOBJ : Flag for linear objective function.

=0 $F(\vec{X})$ is nonlinear function of $(X_1 X_2 \ldots X_{NDV})$

=1 $F(\vec{X})$ is a linear function of $(X_1 X_2 \ldots X_{NDV})$

MAXGSN : Maximum allowable number of golden section iterations. Default=20.

MAXODM : Maximum allowable number of one-dimensional minimizations per unconstrained minimization. Default=6.

MAXRSF : Maximum allowable number of unconstrained minimizations. Default=15.

MFLAG : Flag for penalty multiplier initialization.

=0 ; initial r_p is computed by NEWSUMT.

=1 ; r_p specified by the main program is used as the initial value.

NDV : Number of design variables.

NTCE : Number of constraints considered.

4.2 Analysis Program

The primary subroutine of the analysis program supplied by the user should have the following arguments:

```
SUBROUTINE name (INFO, X, OBJ, DOBJ, G, GB, DG, N1, N2,
N3, N4, RAN, NRANDM, IAN, NIANDM)
```

name : Subroutine name which is identical to the first argument of the CALL NEWSUM statement issued in the main program.

INFO : Control parameter

=1 ; evaluates objective function only.

=2 ; evaluates all constraint functions.

=3 ; evaluates gradient of objective function.

=4 ; evaluates gradients of nonlinear constraint functions.

=5 ; evaluates gradient of linear constraints only.

X(N1) : current design variables.

OBJ	:	objective function value.
DOBJ(N1)	:	gradient of objective function.
DOBJ(N3)	:	second derivatives of objective function.
G(N2)	:	constraint function values.
GB(N2)	:	usually not used.
DG(N4)	:	gradient of constraint functions stored in a vector form.

$$\begin{aligned}
&DG(1)=\partial g_1/\partial X_1\\
&DG(2)=\partial g_2/\partial X_1\\
&\quad\vdots\\
&DG(N2)=\partial g_{N2}/\partial X_1\\
&DG(N2+1)=\partial g_1/\partial X_2\\
&\quad\vdots\\
&DG(N4)=\partial g_{N2}/\partial X_{N1}
\end{aligned}$$

N1	:	maximum number of design variables	Parameters used for variable dimensioning of primary arrays.
N2	:	maximum number of constraints	
N3	:	N1(N1+1)/2	
N4	:	N1*N2	

RAN(NRANDM)	:	real array allocated by the user.
NRANDM	:	dimension of RAN.
IAN(NIANDM)	:	integer array allocated by the user.
NIANDM	:	dimension of IAN.

Fundamental structure of the analysis program which the user must supply is shown in Fig. 2.

4.3 Description of NEWSUMT Subroutines

NEWSUM : Primary subroutine which is called by the user's main program and supervises control of the iteration process.

BLOCK DATA : Initialize the default values of control parameters and internal variables residing in the labeled COMMON blocks.

DIRCTN : Direction finding for unconstrained minimization process by means of modified Newton's method. If finite difference gradient calculation is asked for, gradient of objective and constraint functions are computed in this subroutine. Also automated search for an appropriate transition parameter is carried out whenever necessary.

FUNCTN : Called by ODM (one-dimensional minimization) and evaluates the composite function value for a given design. Quadratic extended penalty function scheme is administered by this subroutine.

ODM : One-dimensional minimization is carried out along the direction provided by DIRCTN by means of the golden section algorithm. Linear constraints are treated separately from nonlinear constraints, since the distances to linear constraint boundaries are readily calculated.

PRINTD : This routine prints debugging information on a line printer. For users who wish to use NEWSUMT as a black box optimizer, this routine may be replaced by a dummary subroutine which does nothing.

RFACTR : Called only once at the beginning to compute the initial penalty multiplier r_p.

SADØØ7 : A part of the linear equation solver used in DIRCTN. A positive definite symmetric matrix stored in a vector form (upper triangular only) is decomposed into a LDL^T form. If a matrix is almost singular or non-positive definite, an error flag is turned on, and the direction finding process is switched to the direction of steepest descent.

SADØØ8 : A part of the linear equation solver used in DIRCTN. Upon successful decomposition of the coefficient matrix of SADØØ7, this routine is called for back and forward substitutions.

V. PRACTICAL CONSIDERATIONS

5.1 Formulation

The standard form given in Eq. (1) is sufficient theoretically, but for numerical stability and efficiency, it is always beneficial to normalize or to scale design variables and constraints. The optimization process tends to be stable when all $\max_j(\partial g_q/\partial X_j)$ have similar orders of magnitude and all $g_q(\vec{X}), q=1,2,\ldots NTCE$, also have similar orders of magnitude. The user should be aware that ill conditioning of the Hessian matrix of ϕ will result in extremely poor performance of this program. For example, consider a case where there is only one active constraint. Without contribution from the objective function, the rank of the matrix J of Eq. 9 is only 1. Haftka proposes to use diagonal perturbation (Ref. 2), but this also fails if the magnitudes of some diagonals are extremely small. Contributions from the objective function tend to alleviate this difficulty, provided they are significant.

5.2 Choice of Control Parameters

The most important parameter which the user should consider is the penalty multiplier decrease ratio (RACUT). If RACUT is very small, then the penalty multiplier RA is decreased very rapidly and experience shows that iterative search tends to be terminated prematurely as the design gets very close to a small number of constraints. If RACUT is large, then convergence will be slow, but the optimal design will usually be of better quality. For difficult problems, RACUT can be as large as 0.5~0.6. For easy problems, RACUT may be taken as 0.1~0.5. MAXRSF and RAMIN should be determined based on the assigned value for RACUT.

The maximum allowable number of one-dimensional minimization per unconstrained minimization should be specified independent of the number of design variables. For most practical problems, the modified Newton's method is very effective and convergence will usually be achieved in three to five one-dimensional minimizations. The maximum number of golden section iterations (MAXGSN) is difficult to estimate, but the default value (20) has been used extensively with satisfactory performance.

5.3 Variable Transfer Strategy

The user may be uncomfortable to see a long list of arguments in SUBROUTINE NEWSUM. For ordinary small applications this looks awkward but when the problem size is large and optimum usage of main memory is required, this transfer strategy becomes beneficial since it allows dynamic array allocation. There are a number of cases where the analysis programs provided by the user also require that certain arrays should be allocated dynamically. The arrays RAN and IAN are available for this purpose. Usually, all the arrays are packed in RAN and IAN sequentially and addressing pointers should be stored as a part of IAN.

VI. EXAMPLES

Two simple examples are included to illustrate the scheme for writing computer programs to interface with NEWSUMT. They may also serve as test cases when NEWSUMT is initially implemented on a particular computer system. Both examples are self contained and need no input data, therefore an executable module created by linking each example program with the NEWSUMT program should readily be processed. Lineprinter outputs are abbreviated because of large volume of data from each iteration, but they should be sufficient to check the results which the user obtains. Even with single precision floating point variables, on a 32 bits/word machine, all results should agree to at least three significant digits with the results given herein. In the printed output, TOTAL FUNCTION stands for the composite function $\phi(\vec{X}, r_p)$ and OBJECTIVE FUNCTION refers to $F(\vec{X})$.

(1) Minimize $F(\vec{X}) = 10X_1 + X_2$

Subject to

$$g_1(\vec{X}) = 2X_1 - X_2 - 1 \geq 0$$
$$g_2(\vec{X}) = X_1 - 2X_2 + 1 \geq 0$$
$$g_3(\vec{X}) = X_1^2 + 2X_1 + 2X_2 - 1 \geq 0$$

This example is solved by a program given on p. 28. The name of the subroutine for analysis is EXAMP1 which must be declared in the main program as an external parameter. All gradient information is expressed analytically and computed in the analysis program as represented by vectors, DOBJ, DDOBJ and DG.

An initial design for this particular example is chosen as (2.0, 1.0), which satisfies all constraints and gives the objective function equal to 21.0. The final result given on page 20 indicates that an optimal design (0.5515, 0.1006) is obtained with corresponding objective function equal to 5.5917 One can easily observe that critical constraints for this optimal design are $g_1(\vec{X})$ and $g_3(\vec{X})$, since their values are 0.3745×10^{-4} and 0.4768×10^{-4} while $g_2(\vec{X})$ is 1.351.

This example is helpful in illustrating the iteration characteristics of the NEWSUMT optimizer. Figure 4 shows the design space and iteration paths using five different initial designs (one feasible, one critical, three infeasible). A conceptually attractive feature of the SUMT algorithm, namely its tendency to "funnel down the middle of the feasible space," is graphically illustrated by this example. It is this tendency of the NEWSUMT program, to generate a sequence of steadily improving noncritical designs, that make it especially well suited to engineering design applications.

(2) Minimize

$$F(\vec{X})=(X_1^2-5X_1)+(X_2^2-5X_2)+(2X_3^2-21X_3)+(X_4^2+7X_4)+50$$

$$g_1(\vec{X})=(-X_1^2-X_1)+(-X_2^2+X_2)+(-X_3^2-X_3)+(-X_4^2+X_4)+8 \geq 0$$

$$g_2(\vec{X})=(-X_1^2+X_1)+(-2X_2^2)+(-X_3^2)+(-2X_4^2+X_4)+10 \geq 0$$

$$g_3(\vec{X})=(-2X_1^2-2X_1)+(-X_2^2+X_2)+(-X_3^2)+X_4+5 \geq 0$$

The optimal point of this well known Rosen-Suzuki problem is

$$F(0.0, 1.0, 2.0, -1.0) = 6.0$$

The main program and the analysis program to solve this problem with NEWSUMT are given on pages 32 and 33, respectively. The analysis subroutine is called EXAMP2, which is declared as an external parameter in the main program. The subroutine EXAMP2 is written so that it can evaluate the analytic derivatives of objective and constraint functions. But in the particular run shown here, the capability to compute all gradients using finite difference is activated by specifying IDF = 5 in the main program, thus DOBJ, DDOBJ and DG are never computed in EXAMP2. The user is encouraged to test the same program using analytic gradients, i.e. simply changing IDF to 0 in the main program.

The initial design is selected as (1.0, 1.0, 1.0, 1.0) which satisfies all constraints and the corresponding objective function value is 31.0. The iteration process converges in 10 stages, yielding an optimal design

$$F(0.0020, 1.000, 1.998, -1.002) = 6.000427$$

which agrees satisfactorily with the theoretical optimal design. Constraints $g_1(\vec{X})$ and $g_2(\vec{X})$ are active since they are respectively 0.1545×10^{-3} and 0.1106×10^{-3} while $g_2(\vec{X})$ is 0.9983.

```
C     ----------------------------------
C     MAIN PROGRAM FOR PROBLEM EXAMPLE 1
C     ----------------------------------
C
C     DECLARE THE NAME OF ANALYSIS PROGRAM AS AN EXTERNAL ENTRY
          EXTERNAL   EXAMP1
C
          DIMENSION  X0(2)   , X(2)     , SN(2)    , G(3)     , DG(6)   ,
         *           DOBJ(2) , DDOBJ(3), S(2)     , DH(2)    , IIK(2)  ,
         *           BU(2)   , BL(2)    , ISIDE(2), GB(3)    , G1(3)   ,
         *           G2(3)   , G3(3)    , ILIN(3) ,
         *           FDCV(2) , RAN(1)   , IAN(1)
          COMMON/CONTRL/C     , EPSGSN, EPSODM, EPSRSF, G0     , P      ,
         *      RA     , RACUT , RAMIN , STEPMX,
         *      IFD    , JPRINT, JSIGNG, LOBJ  , MAXGSN, MAXODM, MAXRSF,
         *      MFLAG  , NDV   , NTCE
C
          N1 = 2
          N2 = 3
          N3 = 3
          N4 = 6
          NDV = 2
          NTCE = 3
          LOBJ = 1
          DO 100 I = 1, NDV
          BL(I) = 0.0
          ILIN(I) = 1
      100 ISIDE(I) = 1
          ILIN(3) = 0
          NRANDM = 1
          NIANDM = 1
          MFLAG  = 0
          JPRINT = 1
C
C     SPECIFY THE INITIAL DESIGN VARIABLES
          X0(1) = 2.0
          X0(2) = 1.0
C
          CALL NEWSUM(EXAMP1,
         *    BL    , BU    , DDOBJ , DG    , DH     , DOBJ  ,
         *    FDCV  , FMIN  , G     , GB    , G1     , G2    , G3     ,
         *    OBJ   , OBJMIN, S     , SN    , X      , X0    ,
         *    IIK   , ILIN  , ISIDE , N1    , N2     , N3    , N4     ,
         *    RAN   , NRANDM, IAN   , NIANDM )
C
          STOP
          END
```

Objective function is linear.

Only lower bounds of 0.0 are imposed to x_1 and x_2.

g_1 and g_2 are linear, g_3 are nonlinear.

```
C   --------------------------------------
C   ANALYSIS PROGRAM FOR EXAMPLE PROBLEM 1
C   --------------------------------------
      SUBROUTINE EXAMP1(INFO  , X     , OBJ   , DOBJ  , DDOBJ , G     ,
     *                  GB    , DG    , N1    , N2    , N3    , N4    ,
     *                  RAN   , NRANDM, IAN   , NIANDM )
C
      DIMENSION X(N1)       , DOBJ(N1)    , DDOBJ(N3) , G(N2)       ,
     *          DG(N4)      , GB(N2)      , RAN(NRANDM), IAN(NIANDM)
C
      A = X(1)
      B = X(2)
C
      GO TO (100, 200, 300, 400, 500), INFO
C
C  EVALUATE OBJECTIVE FUNCTION
  100 OBJ = 10.0*A + B
      RETURN
C
C  EVALUATE CONSTRAINT FUCTIONS
  200 G(1) = 2.0*A - B - 1.0
      G(2) = A - 2.0*B + 1.0
      G(3) = -A*A + 2.0*(A+B) - 1.0
      RETURN
C
C  EVALUATE THE FIRST AND THE SECOND ORDER DERIVATIVES OF
C  THE OBJECTIVE FUNCTION
  300 DOBJ(1) = 10.0
      DOBJ(2) = 1.0
      DDOBJ(1) = 0.0
      DDOBJ(2) = 0.0
      DDOBJ(3) = 0.0
      RETURN
C
C  EVALUATE GRADIENT OF NONLINEAR CONSTRAINTS
  400 DG(3) = -2.0*A + 2.0
      DG(6) = 2.0
      RETURN
C
C  EVALUATE GRADIENT OF LINEAR CONSTRAINTS
  500 DG(1) = 2.0
      DG(2) = 1.0
      DG(4) = -1.0
      DG(5) = -2.0
      RETURN
      END
```

Note: IFD=0, i.e. analytical derivatives for all functions are computed in the user's analysis program, is the default option.

Example Problem 1 - Abridged output (JPRINT=1)

```
*********************    N E W S U M T   O P T I M I Z E R    *********************

                                CONTROL PARAMETERS
INITIAL TRANSITION POINT . . . . . . . . . . . GO =    0.1000E+00
TRANSITION POINT EXPONENT . . . . . . . . . . . P =    0.5000E+00
INITIAL TRANSITION POINT COEFFICIENT . . . . . C =    0.2000E+00
GOLDEN SECTION CONVERGENCE . . . . . . . . EPSGSN =    0.1000E-02
UNCONSTRAINED MINIMIZATION CONVERGENCE EPSODM =    0.1000E-02
CONVERGENCE AMONG RESPONSE SURFACES     EPSRSF =    0.5000E-03
RESPONSE FACTOR REDUCTION  RATIO ...... RACUT =    0.1000E+00
MINIMUM ALLOWABLE RESPONSE FACTOR ..... IRAMIN=    0.1000E-12
MAXIMUM ALLOWABLESTEP SIZE ........... STEPMX =    0.1000E+11
MAXIMUM ALLOWABLE GOLDEN SECTIONS       MAXGSN =    20
MAXIMUM NUMBER OD O.D.M. PER SURFACE    MAXODM =     6
MAXIMUM ALLOWABLE RESPONSE SURFACES     MAXRSF =    30
PRINTOUT CONTROL.......................JPRINT =     1
FINITE DIFFERENCE GRADIENT CONTROL......   IFD =     0
```

Analytical derivatives

```
                                SYSTEM PARAMETERS
NUMBER OF DESIGN VARIABLES............. .. NDV =     2
NUMBER OF EFFECTIVE CONSTRAINTS ........ NTCE =     3


   INITIAL DESIGN ANALYSIS SUMMARY
   -------------------------------
        INITIAL DESIGN VARIABLE VECTOR
          0.2000E+01   0.1000E+01
        SIDE CONSTRAINTS
                -1            -2
          0.2000E+01   0.1000E+01
        CONSTRAINTS
          0.2000E+01   0.1000E+01   0.1000E+01
        OBJECTIVE FUNCTION =    0.2100000E+02
```

x_1 and x_2

Lower bound side constraint for x_2 is 1.000.

(If the index is positive, it indicates upper bound side constraint.)

g_1, g_2 and g_3

$F(\vec{X})$

One dimensional search 2 is restarted using complete analyses, since the usage of quadratic approximation for each constraint resulted in higher $\Phi(X, r_p)$ than its value at the beginning of this search.

r_p

```
----------------------------------------
OPTIMIZATION OF RESPONSE SURFACE NO.  1     PENALTY MULTIPLIER =    0.525000E+01
----------------------------------------
   ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 0.399643E+02  OBJECTIVE FUNCTION= 0.165580E+02
   ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 0.398529E+02  OBJECTIVE FUNCTION= 0.162391E+02

   REPEAT ODM BY COMPLETE ANALYSIS
   ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 0.398509E+02  OBJECTIVE FUNCTION= 0.163712E+02
   ONE DIMENSIONAL SEARCH 4  TOTAL FUNCTION= 0.398347E+02  OBJECTIVE FUNCTION= 0.163389E+02

   RESULTS AT THE END OF THIS UNCONSTRAINED MINIMIZATION
        INITIAL DESIGN VARIABLE VECTOR
          0.1557E+01   0.7895E+00
        SIDE CONSTRAINTS
               -1           -2
          0.1557E+01   0.7895E+00
        CONSTRAINTS
          0.1322E+01   0.9786E+00   0.1269E+01
        TOTAL FUNCTION      =   0.3983470E+02
        OBJECTIVE FUNCTION =   0.1633887E+02
```

$\Phi(\vec{X}, r_p)$

$F(\vec{X})$

```
---------------------------------------

 OPTIMIZATION OF RESPONSE SURFACE NO.  2     PENALTY MULTIPLIER =   0.525000E+00

---------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 0.139056E+02  OBJECTIVE FUNCTION= 0.945919E+01
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 0.136870E+02  OBJECTIVE FUNCTION= 0.887688E+01

    REPEAT ODM BY COMPLETE ANALYSIS
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 0.136863E+02  OBJECTIVE FUNCTION= 0.891434E+01
    ONE DIMENSIONAL SEARCH 4  TOTAL FUNCTION= 0.136863E+02  OBJECTIVE FUNCTION= 0.891775E+01

    RESULTS AT THE END OF THIS UNCONSTRAINED MINIMIZATION
         INITIAL DESIGN VARIABLE VECTOR
           0.8574E+00   0.3626E+00
         SIDE CONSTRAINTS
                -1           -2
           0.8574E+00   0.3626E+00
         CONSTRAINTS
           0.3499E+00   0.1133E+01   0.7049E+00
         TOTAL FUNCTION       =   0.1368631E+02
         OBJECTIVE FUNCTION =   0.8917752E+01




---------------------------------------

 OPTIMIZATION OF RESPONSE SURFACE NO. 10     PENALTY MULTIPLIER =   0.525000E-08

---------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 0.559644E+01  OBJECTIVE FUNCTION= 0.559623E+01
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 0.559643E+01  OBJECTIVE FUNCTION= 0.559616E+01
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 0.559642E+01  OBJECTIVE FUNCTION= 0.559617E+01

    RESULTS AT THE END OF THIS UNCONSTRAINED MINIMIZATION
         INITIAL DESIGN VARIABLE VECTOR
           0.5515E+00   0.1006E+00
         SIDE CONSTRAINTS
                -1           -2
           0.5515E+00   0.1006E+00
         CONSTRAINTS
           0.3745E-04   0.1351E+01   0.4768E-04      Optimal design
         TOTAL FUNCTION       =   0.5596422E+01        x1 = 0.5515    x2 = 0.1006
         OBJECTIVE FUNCTION =   0.5596172E+01
                                                       F(x1, x2) = 5.591672
    FINAL RESULTS OF OPTIMIZATION
    ----------------------------
         CURRENT DESIGN VARIABLE VECTOR
           0.5515E+00   0.1006E+00
         SIDE CONSTRAINTS
                -1           -2                        No side constraints are critical.
           0.5515E+00   0.1006E+00
         CONSTRAINTS                                   g1 and g3 are critical constraints
           0.3745E-04   0.1351E+01   0.4768E-04         while g3 are not.
         TOTAL FUNCTION       =   0.5596422E+01
         OBJECTIVE FUNCTION =   0.5596172E+01

         FINAL STATISTICS
         ----------------
              NUMBER OF RESPONSE SURFACE . . . .     10
              NUMBER OF ONE DIMENSIONAL SEARCH .     34
              NUMBER OF ANALYSES

                   OBJECTIVE FUNCTION . . . . . . . . . . . . . . .  309
                   GRADIENT OF OBJECTIVE FUNCTION . . . . . . . . .    1
                   CONSTRAINT FUNCTIONS . . . . . . . . . . . . . .  110
                   GRADIENT OF LINEAR CONSTRAINT FUNCTIONS . . . .     1
                   GRADIENT OF NONLINEAR CONSTRAINT FUNCTIONS . .     34    110+199 = 309
                   APPROXIMATE CONSTRAINT FUNCTIONS . . . . . . .    199
```

```
C   ----------------------------------
C   MAIN PROGRAM FOR PROBLEM EXAMPLE 2
C   ----------------------------------
C
C   DECLARE THE NAME OF ANALYSIS PROGRAM AS EXTERNAL ENTRY
      EXTERNAL  EXAMP2
C
      DIMENSION X0(4)     , X(4)       , SN(4)     , G(3)      ,
     *          DG(12)    , DOBJ(4)    , DDOBJ(10), S(4)       ,
     *          DH(4)     , IIK(4)     , BU(4)     , BL(4)     ,
     *          ISIDE(4)  , GB(3)      , ILIN(3)   , FDCV(4)   ,
     *          G1(3)     , G2(3)      , G3(3)     , XB(4)     ,
     *          RAN(1)    , IAN(1)
C
      COMMON/CONTRL/ C      , EPSGSN, EPSODM, EPSRSF, G0     , P
     *     , RA     ,  RACUT , RAMIN , STEPMX
     *     , IFD    , JPRINT , JSIGNG, LOBJ   , MAXGSN, MAXODM, MAXRSF
     *     , MFLAG  , NDV    , NTCE
C
C   INITIALIZE NON-DEFAULT CONTROL PARAMETERS
      NRANDM = 1
      NIANDM = 1
      N1 = 4
      N2 = 3
      N3 = ((N1+1) * N1)/2
      N4 = N1 * N2
      NDV = 4
      NTCE = 3
      LOBJ = 0
      MFLAG = 0
      DO 100 I = 1, NDV
  100 ISIDE(I) = 0
      DO 110 I = 1, NTCE
  110 ILIN(I) = 0
      IFD = 5
      JPRINT = 2
C
C   INITIALIZE THE STARTING DESIGN
      DO 120 I = 1, NDV
  120 X0(I) = 1.0
C
      CALL NEWSUM(EXAMP2
     *     , BL     , BU     , DDOBJ , DG     , DH     , DOBJ
     *     , FDCV   , FMIN   , G     , GB     , G1     , G2     , G3
     *     , OBJ    , OBJMIN, S      , SN     , X      , X0
     *     , IIK    , ILIN   , ISIDE , N1     , N2     , N3     , N4
     *     , RAN    , NRANDM, IAN    , NIANDM)
C
      STOP
      END
```

All variables are unbounded. (No side constraints)

All constraints and objective functions are nonlinear.

Finite difference scheme is activated to compute derivatives of all nonlinear functions.

```
C  ---------------------------------------
C  ANALYSIS PROGRAM FOR EXAMPLE PROBLEM 2
C  ---------------------------------------
C
      SUBROUTINE EXAMP2(INFO   , X      , OBJ    , DOBJ   , DDOBJ  ,
     *            G      , GB     , DG     , N1     , N2     , N3     , N4     ,
     *            RAN    , NRANDM, IAN     , NIANDM)
C
      DIMENSION X(4)      , DOBJ(4)     , DDOBJ(10), G(3)        ,
     *          DG(12)    , GB(3)       , RAN(NRANDM), IAN(NIANDM)
C
      T = X(1)
      U = X(2)
      V = X(3)
      W = X(4)
C
      GO TO (100, 200, 300, 400, 500), INFO
C
C  EVALUATE OBJECTIVE FUNCTION
  100 OBJ  = T*T - 5.0*T + U*U - 5.0*U + 2.0*V*V -21.0*V + W*W
     *         + 7.0*W + 50.0
      RETURN
C
C  EVALUATE CONSTRAINT FUNCTIONS
  200 CONTINUE
      G(1) = -T*T - T - U*U + U - V*V - V - W*W + W + 8.0
      G(2) = -T*T + T - 2.0*U*U - V*V - 2.0*W*W + W + 10.0
      G(3) = -2.0*T*T - 2.0*T - U*U + U - V*V + W + 5.0
      RETURN
C
C  EVALUATE THE FIRST AND THE SECOND ORDER DERIVATIVES OF OBJECTIVE
C  FUNCTION
  300 CONTINUE
      DOBJ(1) = 2.0*T - 5.0
      DOBJ(2) = 2.0*U - 5.0
      DOBJ(3) = 4.0*V - 21.0
      DOBJ(4) = 2.0*W + 7.0
C
      DO 310 I = 1, 10
  310 DDOBJ(I) = 0.0
      DDOBJ(1) = 2.0
      DDOBJ(3) = 2.0
      DDOBJ(6) = 4.0
      DDOBJ(10) = 2.0
      RETURN
C
C  EVALUATE GRADIENT OF NONLINEAR CONSTRAINTS
  400 CONTINUE
      DG(1) = -2.0*T - 1.0
      DG(2) = -2.0*T + 1.0
      DG(3) = -4.0*T - 2.0
      DG(4) = -2.0*U + 1.0
      DG(5) = -4.0*U
      DG(6) = -2.0*U + 1.0
      DG(7) = -2.0*V - 1.0
      DG(8) = -2.0*V
      DG(9) = -2.0*V
      DG(10) = -2.0*W + 1.0
      DG(11) = -4.0*W + 1.0
      DG(12) = 1.0
      RETURN
C
C  EVALUATE GRADIENT OF LINEAR CONSTRAINTS
  500 CONTINUE
      RETURN
      END
```

Note: This part of the program is not used, since IFD=5 is specified in the MAIN program.

Example Problem 2 - Abridged Output (JPRINT=2)

```
*********************   N E W S U M T   O P T I M I Z E R   *********************

                              CONTROL PARAMETERS
  INITIAL TRANSITION POINT . . . . . . . . . . . G0 =   0.1000E+00
  TRANSITION POINT EXPONENT . . . . . . . . . . . P =   0.5000E+00
  INITIAL TRANSITION POINT COEFFICIENT . . .    C =   0.2000E+00
  GOLDEN SECTION CONVERGENCE . . . . .   . EPSGSN =   0.1000E-02
  UNCONSTRAINED MINIMIZATION CONVERGENCE EPSODM =   0.1000E-02
  CONVERGENCE AMONG RESPONSE SURFACES    EPSRSF =   0.5000E-03
  RESPONSE FACTOR REDUCTION  RATIO ...... RACUT =   0.1000E+00
  MINIMUM ALLOWABLE RESPONSE FACTOR ..... IRAMIN=   0.1000E-12
  MAXIMUM ALLOWABLESTEP SIZE ........... STEPMX =   0.1000E+11
  MAXIMUM ALLOWABLE GOLDEN SECTIONS      MAXGSN =    20
  MAXIMUM NUMBER OD O.D.M. PER SURFACE   MAXODM =     6
  MAXIMUM ALLOWABLE RESPONSE SURFACES    MAXRSF =    30
  PRINTOUT CONTROL.......................JPRINT =     2
  FINITE DIFFERENCE GRADIENT CONTROL......  IFD =     5

                              SYSTEM PARAMETERS
  NUMBER OF DESIGN VARIABLES............. .. NDV =     4
  NUMBER OF EFFECTIVE CONSTRAINTS ........ NTCE =     3


     INITIAL DESIGN ANALYSIS SUMMARY
     -------------------------------
          INITIAL DESIGN VARIABLE VECTOR
            0.1000E+01  0.1000E+01  0.1000E+01  0.1000E+01
          CONSTRAINTS
            0.4000E+01  0.6000E+01  0.1000E+01
          OBJECTIVE FUNCTION =   0.3100000E+02


------------------------------------------

 OPTIMIZATION OF RESPONSE SURFACE NO.  1      PENALTY MULTIPLIER =    0.218824E+02
------------------------------------------

----- DIRECTION FINDING -----
      TRANSITION POINT =    0.100000E+00  ——— ε

      DIRECTION COMPUTED BY MODIFIED NEWTON'S METHOD
      SLOPE = -0.2264E+02  ——— S^T·∇Φ
      NORMALIZED DIRECTION VECTOR
       -0.2707E+00 -0.2924E+00  0.3743E+00 -0.8374E+00  ——— S^T

----- ONE DIMENSIONAL MINIMIZATION RUN NO.   1 -----  ——— α_min
----- END OF O.D.M. DISTANCE FOR MIN. PT. =  0.9407793E+00 -----
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 0.513895E+02  OBJECTIVE FUNCTION= 0.205227E+02

      NOT CONVERGED - CHECK1= -0.1000E+01  CHECK2=  0.1946E+19  EPSODM=  0.1000E-02

----- DIRECTION FINDING -----
      TRANSITION POINT =   0.425813E+00

      DIRECTION COMPUTED BY MODIFIED NEWTON'S METHOD
      SLOPE = -0.1841E+02
      NORMALIZED DIRECTION VECTOR
       -0.3202E+00  0.9768E-01  0.3788E-01 -0.9416E+00

----- ONE DIMENSIONAL MINIMIZATION RUN NO.   2 -----
----- END OF O.D.M. DISTANCE FOR MIN. PT. =  0.8987486E+00 -----
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 0.434400E+02  OBJECTIVE FUNCTION= 0.152161E+02
```

```
                    .
                    .
                    .
-----------------------------------------

 OPTIMIZATION OF RESPONSE SURFACE NO. 10     PENALTY MULTIPLIER =   0.218824E-07

-----------------------------------------

----- DIRECTION FINDING -----
          TRANSITION POINT =    0.912702E-05

          DIRECTION COMPUTED BY MODIFIED NEWTON'S METHOD
          SLOPE = -0.9156E+01
          NORMALIZED DIRECTION VECTOR
            0.4763E+00  0.7524E+00  0.3046E+00 -0.3379E+00

----- ONE DIMENSIONAL MINIMIZATION RUN NO.    1 -----
----- END OF O.D.M. DISTANCE FOR MIN. PT. =  0.6073422E-04 -----
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 0.600079E+01  OBJECTIVE FUNCTION= 0.600052E+01
                    .
                    .
                    .

----- ONE DIMENSIONAL MINIMIZATION RUN NO.    3 -----

    REPEAT ODM BY COMPLETE ANALYSIS
----- END OF O.D.M. DISTANCE FOR MIN. PT. =  0.1752282E-05 -----
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 0.600077E+01  OBJECTIVE FUNCTION= 0.600043E+01

         CONVERGED - CHECK1=  0.2861E-05  CHECK2=  0.9537E-06  EPSODM=  0.1000E-02

    RESULTS AT THE END OF THIS UNCONSTRAINED MINIMIZATION
         INITIAL DESIGN VARIABLE VECTOR
           0.2004E-02  0.1000E+01  0.1998E+01 -0.1002E+01
         CONSTRAINTS
           0.1545E-03  0.9983E+00  0.1106E-03
         TOTAL FUNCTION      =  0.6000766E+01
         OBJECTIVE FUNCTION  =  0.6000427E+01
```

$\vec{X}_{10}$

$(F(\vec{X}_8) - F(\vec{X}_9)) / F(\vec{X}_8)$

```
       CONVERGED - CHECK3=  0.4368E-03  CHECK4=  0.1163E-03  EPSRSF=  0.5000E-03
```

$(F(\vec{X}_{10}) - F(\vec{X}_9)) / F(\vec{X}_9)$

```
       DOUBLE CONVERGENCE CRITERIA IS SATISFIED
```

Both CHECK3 and CHECK4 are smaller than EPSRSF.

```
  FINAL RESULTS OF OPTIMIZATION
  -----------------------------
       CURRENT DESIGN VARIABLE VECTOR
         0.2004E-02  0.1000E+01  0.1998E+01 -0.1002E+01
       CONSTRAINTS
         0.1545E-03  0.9983E+00  0.1106E-03
       TOTAL FUNCTION      =  0.6000766E+01
       OBJECTIVE FUNCTION  =  0.6000427E+01

       FINAL STATISTICS
       ----------------
            NUMBER OF RESPONSE SURFACE . . . .   10
            NUMBER OF ONE DIMENSIONAL SEARCH .   35
            NUMBER OF ANALYSES

                 OBJECTIVE FUNCTION . . . . . . . . . . . . .  699
                 GRADIENT OF OBJECTIVE FUNCTION . . . . . . . .  0
                 CONSTRAINT FUNCTIONS . . . . . . . . . . . .  285
                 GRADIENT OF LINEAR CONSTRAINT FUNCTIONS . . .   0
                 GRADIENT OF NONLINEAR CONSTRAINT FUNCTIONS .    0
                 APPROXIMATE CONSTRAINT FUNCTIONS . . . . . .  274
```

REFERENCES

1. Fiacco, A.V. and McCormick, G.P., "Nonlinear Programming: Sequential Unconstrained Minimization Techniques," John Wiley and Sons, Inc., 1968.

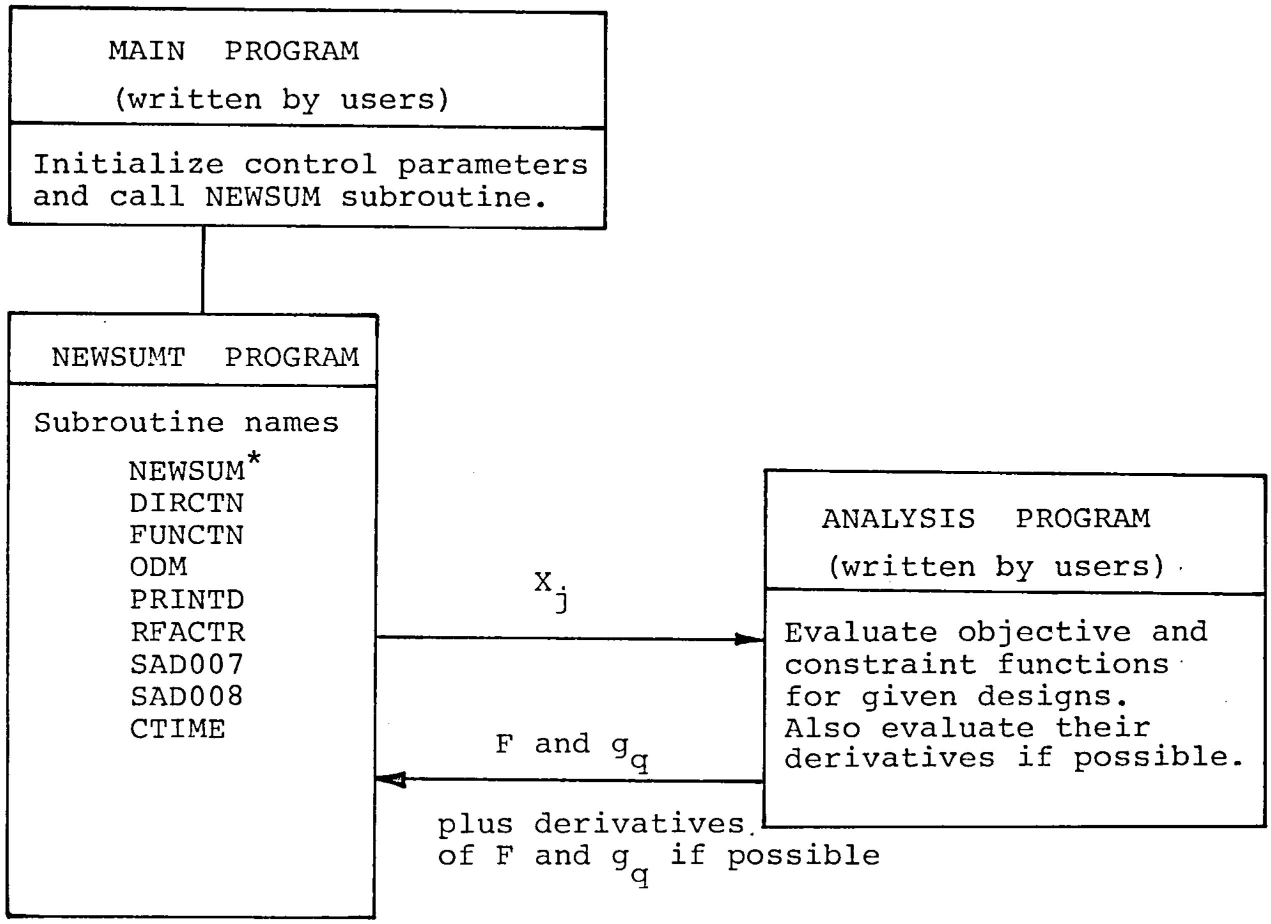
2. Haftka, R., "Automated Procedure for Design of Wing Structures to Satisfy Strength and Flutter Requirements," NASA TN D-7264, 1973.

3. Schmit, L.A. and Miura, H., "An Advanced Structural Analysis/Synthesis Capability - ACCESS 2," Int. J. Num. Meth. Engrg., Vol. 12, No. 2, February 1978, pp. 353-377.

4. Kavlie, D. and Moe, J., "Automated Design of Frame Structures," J. of the Structural Div., ASCE, Vol. 97, No. ST1, January 1971, pp. 33-62.

5. Haftka, R. and Starnes, J.H., "Applications of a Quadratic Extended Interior Penalty Function for Structural Optimization," AIAA Journal, Vol. 14, No. 6, June 1976, pp. 718-724.

6. Cassis, J. and Schmit, L.A., "On Implementation of the Extended Interior Penalty Function," Int. J. Num. Meth. Engrg., Vol. 10, No. 1, January 1976, pp. 3-23.



* Only this subroutine is called by the user's main program to activate the NEWSUMT program.

Fig. 1 Basic Program Organization

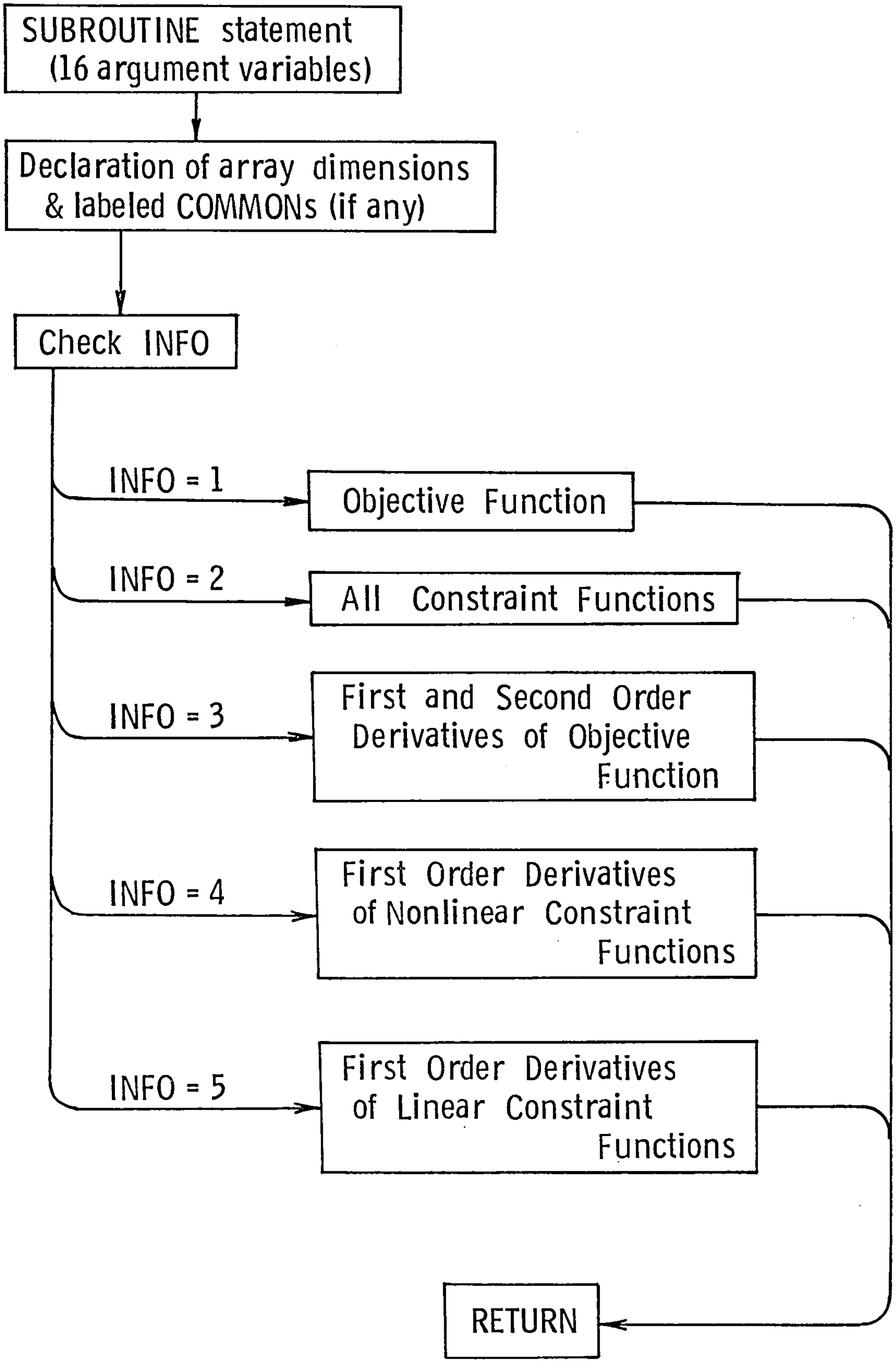


Fig. 2 Structure of the Analysis Program to Evaluate All Functions and Their Derivatives (if available)

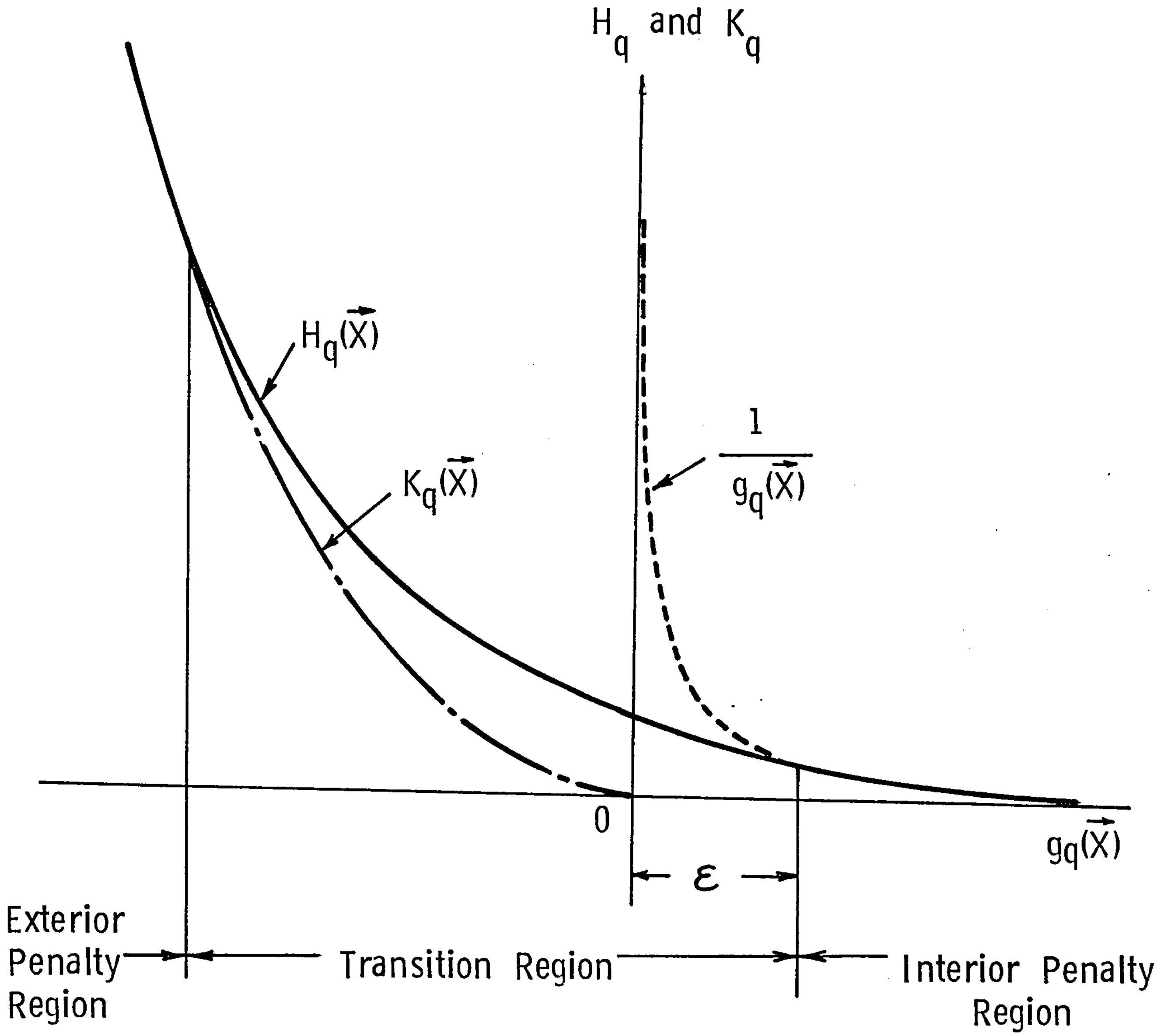


Fig. 3 Integration of Interior-Exterior Penalty Philosophy

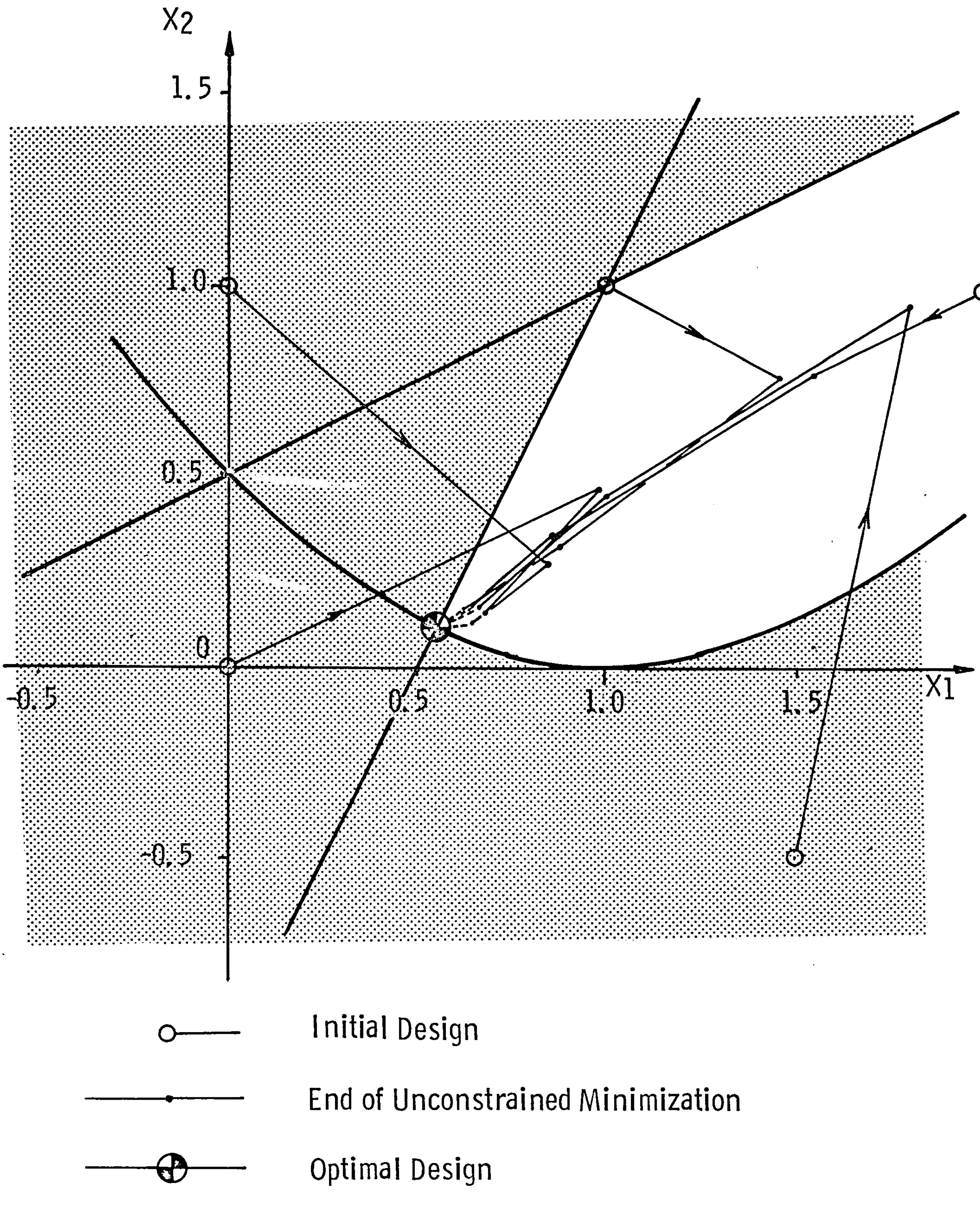
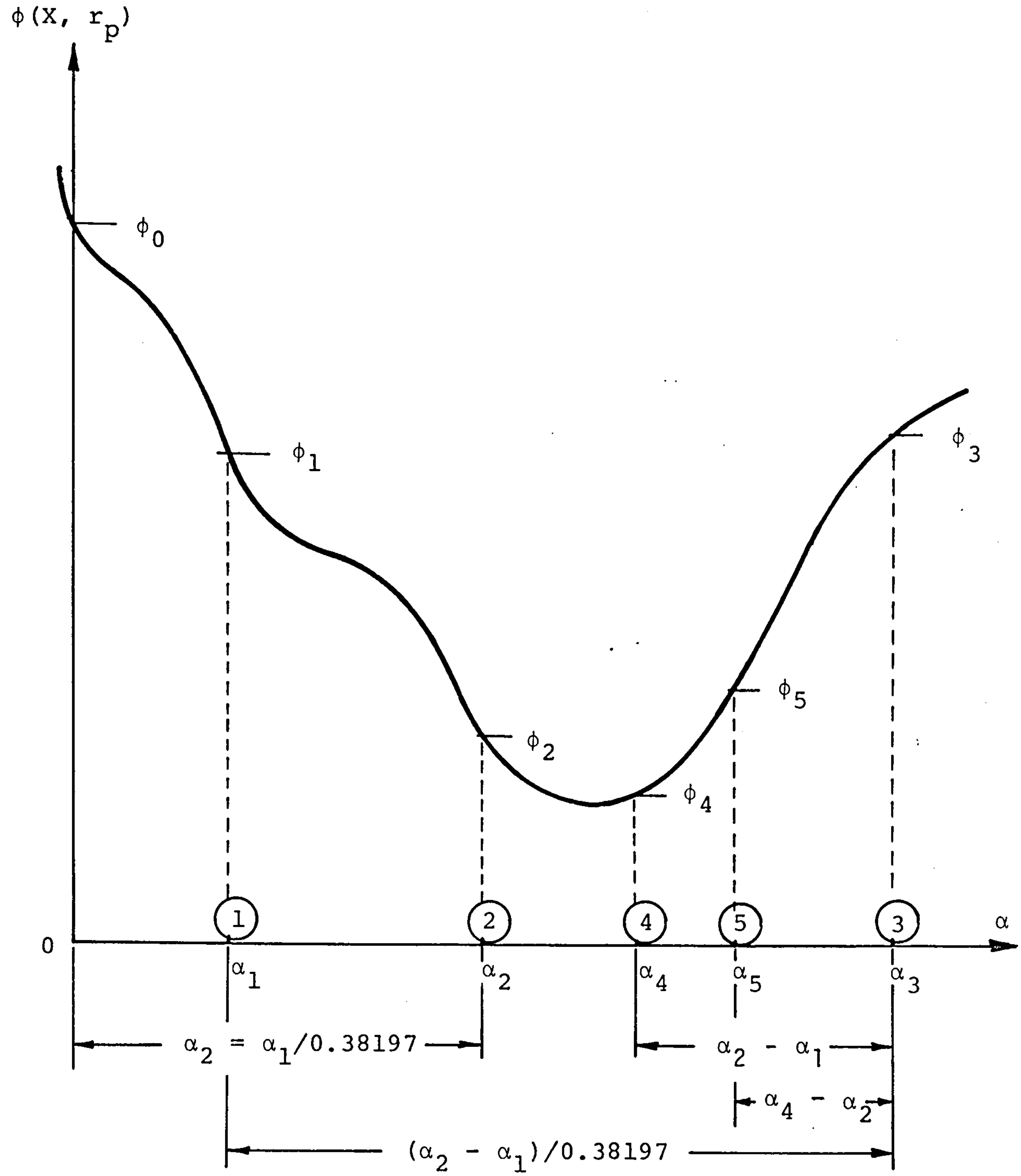


Fig. 4 Iteration Trajectories for Example 1.



Note: Golden section algorithm is initiated using the function values evaluated at α_2, α_4, α_5, & α_3.

Fig. 5 One Dimensional Search Scheme

APPENDIX A

An Interpretation of Quadratic Extended Penalty Functions

The NEWSUMT program is written based on the philosophy of interior penalty functions. However, implementation of the quadratic extended penalty function has made it possible to regard this program as an integration of interior and exterior penalty function philosophy. Let the composite function of an exterior penalty function be defined as

$$\phi_{EX}(\vec{X},R_p) = F(\vec{X}) + R_p \sum_{q=1}^{Q} K_q(\vec{X}) \tag{A1}$$

where

$$K_q(\vec{X}) = \begin{cases} 0 & \text{if } g_q(\vec{X}) \geq 0 \\ [g_q(\vec{X})]^2 & \text{if } g_q(\vec{X}) < 0 \end{cases} \tag{A2}$$

Now compare this with a simplified version of Eqs. (3) and (4)

$$\Phi_{IN}(\vec{X},r_p) = F(\vec{X}) + r_p \sum_{q=1}^{Q} H_q(\vec{X}) \tag{A3}$$

$$H_q(\vec{X}) = \begin{cases} \dfrac{1}{g_q(\vec{X})} & g_q(\vec{X}) \geqq \varepsilon > 0 \\ \dfrac{1}{\varepsilon}\left[\left(\dfrac{g_q(\vec{X})}{\varepsilon}\right)^2 - \dfrac{3g_q(X)}{\varepsilon} + 3\right] & g_q(\vec{X}) < \varepsilon \end{cases} \tag{A4}$$

If a constraint is violated significantly, the dominant term in $H_q(\vec{X})$ is clearly the first term

$$H_q(\vec{X}) \cong [g_q(\vec{X})]^2/\varepsilon^3 \tag{A5}$$

Recalling Eq. (7) for the relation $\varepsilon = C\sqrt{r_p}$

$$H_q(\vec{X}) \cong [g_q(\vec{X})]^2/C^3 r_p^{3/2} \tag{A6}$$

Therefore, if R_p in Eq. (A1) is chosen as

$$R_p = 1/(C^3 r_p^{1/2}) \tag{A7}$$

then the relation given by Eq. (A4) may be interpreted so that it includes the basic ingredients of Eq. (A2). This is illustrated in Fig. 3. It is well recognized that both interior and exterior penalty functions exhibit poor numerical behaviors at or near the constraint boundaries where $g_q(\vec{X})=0$. The quadratic extended penalty function interpolates the penalty function adequately in this critical region.

APPENDIX B

CPU Timing Routine

The function CTIME(I) is used to measure CPU time spent for various parts of data processing in the NEWSUMT program. Timing routines are usually installation dependent and CTIME(I) is available only on IBM 360/91 at UCLA. The function CTIME(I) gives the remaining CPU time, which is the difference between the estimated maximum CPU time specified in the JOB card and the CPU time already spent on the particular job.

For most CDC computers, a function SECOND does a similar job, but it gives the CPU time expended by the particular job. Therefore, CDC users may add a simple function such as

```
FUNCTION CTIME(I)
CALL SECOND(T)
CTIME=-T
RETURN
END.
```

1. Report No. NASA CR-159070	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle NEWSUMT - A Fortran Program for Inequality Constrained Function Minimization - Users Guide		5. Report Date June 1979
		6. Performing Organization Code
7. Author(s) Hirokazu Miura and Lucien A. Schmit, Jr.		8. Performing Organization Report No.
		10. Work Unit No. 505-02-13-42
9. Performing Organization Name and Address University of California, Los Angeles Los Angeles, CA 90024		11. Contract or Grant No. NGR 05-007-337
		13. Type of Report and Period Covered Contractor Report
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Army Project No.

15. Supplementary Notes

W. Jefferson Stroud, Structures and Dynamics Division, Langley Research Center was technical monitor on this grant.

16. Abstract

This report serves as a user's guide for the NEWSUMT computer program. NEWSUMT is a general purpose numerical optimization program for finding the values of $\vec{X}(\vec{X} = X_1, X_2, \ldots X_n)$ that minimize a function $F(\vec{X})$ subject to inequality constraints $g_q(\vec{X}) \geqq 0$, $q = 1,2,\ldots Q$. The user supplies subroutines that define the optimization problem to be solved.

17. Key Words (Suggested by Author(s))		18. Distribution Statement	
Mathematical Programming Numerical Optimization SUMT		Unclassified - Unlimited Subject Category 61	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 46	22. Price* $4.50

*For sale by the National Technical Information Service, Springfield, Virginia 22161

DO NOT REMOVE SLIP FROM MATERIAL

Delete your name from this slip when returning material to the library.

NAME	MS
~~K Belvin (or) Yamaguchi~~	~~230~~
~~Howard University~~	~~[illegible]~~

NASA Langley (Rev. May 1988)

RIAD N-75