NASA Technical Memorandum 4537

CometBoards Users Manual

Release 1.0

James D. Guptill, Rula M. Coroneos, Surya N. Patnaik,
Dale A. Hopkins, and Laszlo Berke

OCTOBER 1996

National Aeronautics and
Space Administration

NASA Technical Memorandum 4537

CometBoards Users Manual

Release 1.0

James D. Guptill and Rula M. Coroneos
Lewis Research Center
Cleveland, Ohio

Surya N. Patnaik
Ohio Aerospace Institute
Brook Park, Ohio

Dale A. Hopkins and Lazlo Berke
Lewis Research Center
Cleveland, Ohio

National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1996

Contents

List of Figures

Figure Page

List of Tables

Section 1
Introduction

Several nonlinear mathematical programming algorithms for structural design applications are available at present. These include the sequence of unconstrained minimizations technique, the method of feasible directions, and the sequential quadratic programming technique. The optimality criteria technique and the fully utilized design concept are two other structural design methods. A project was undertaken to bring all these design methods under a common computing environment so that a designer can select any one of these tools that may be suitable for his/her application. To facilitate selection of a design algorithm, to validate and check out the computer code, and to ascertain the relative merits of the design tools, modest finite element structural analysis programs based on the concept of stiffness and integrated force methods have been coupled to each design method. The code, which contains both design and analysis tools, by reading input information from analysis and design data files, casts the design of a structure as a minimum-weight optimization problem. The code can then solve it by using a specified optimization technique and a specified analysis method. This design code is called CometBoards.

CometBoards is an acronym for “Comparative Evaluation Test Bed of Optimization and Analysis Routines for the Design of Structures.” The term “test bed” is used to emphasize the intended purpose, which is to check out each of the several optimization techniques in a test environment rather than to provide a complete production design system. The term “comparative evaluation” describes its use for determining the relative merits of these techniques as design tools. The organization of the code CometBoards, which is being developed at NASA Lewis Research Center, is shown in figure 1. The key components of CometBoards are the optimizers, the analyzers, the data files, and the central processor or interface module.

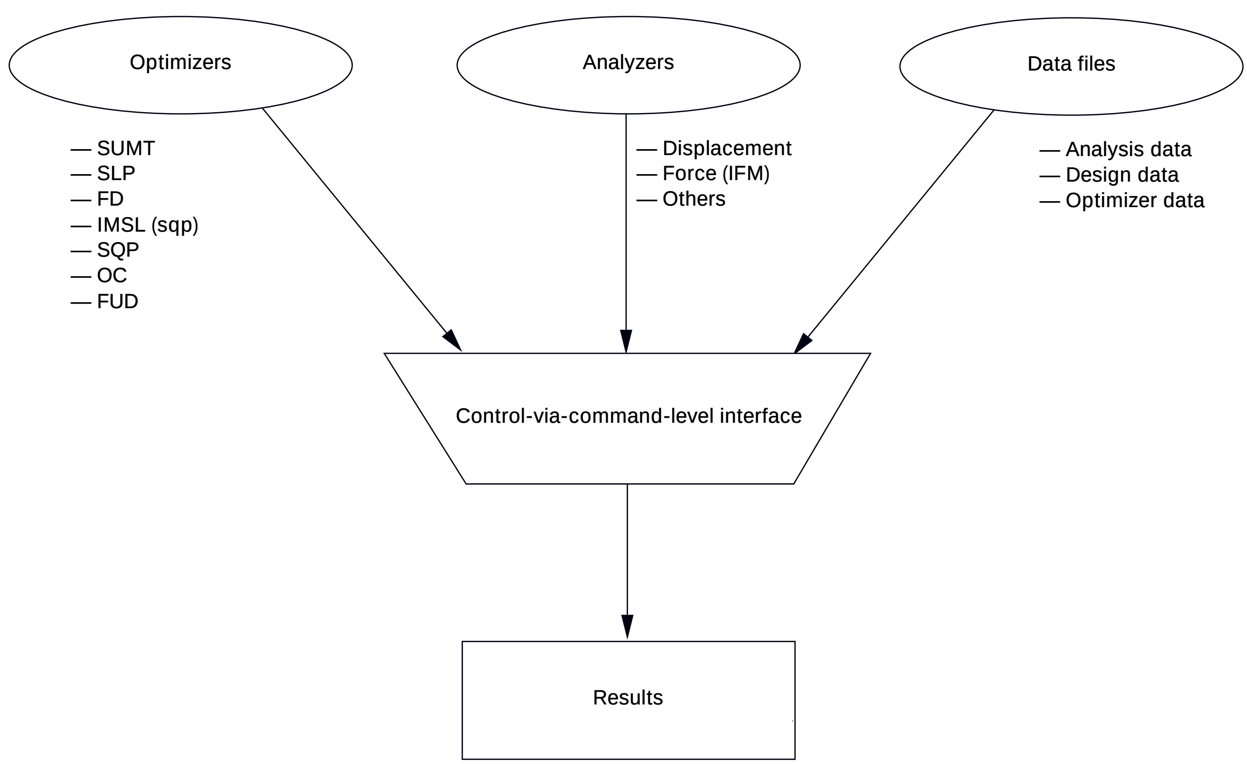


Figure 1.—Organization of CometBoards.

The optimizer module contains seven design algorithms:

(1) The sequence of unconstrained minimizations technique (SUMT), reference 1
(2) The sequential linear programming technique (SLP), reference 2
(3) The modified method of feasible directions (FD), reference 2
(4) The sequential quadratic programming technique from IMSL, Inc., Fortran subroutines (IMSL), reference 3
(5) The sequential quadratic programming (SQP) technique, reference 4
(6) Optimality criteria (OC) methods, reference 5
(7) The fully utilized design (FUD) technique, reference 6

Modest finite element analysis tools available in the analysis module are

(1) A finite element stiffness (displacement) code, ANALYZE/DANLYZE, developed at Wright-Patterson Air Force Base in 1978, which is the primary analyzer, reference 7
(2) The integrated force method (IFM), reference 8
(3) Three versions of the simplified integrated force method (SIFMSD, SIFMD, and SIFMS), reference 9
(4) A closed-form IFM solution that is used to check the other analyzers

The analyzers are modest research-level codes, and their primary purpose is to access and check out the optimizers.

The data files module reads finite element analysis input in the analysis data file; design variables and their groupings plus constraint specifications and their limitations in the design data file; and data specific to optimization algorithms, such as convergence criteria, iteration limits, and constraint thickness, in the optimization data file. The central processor's control-via-command-level interface establishes links between the optimizer, analyzer, and data file modules. It casts the design as an optimization problem and then solves it by using a user-specified analyzer and a user-specified optimizer and stores the solution in an output device. CometBoards is available for Posix-based Cray and Convex computers, Iris and Sun workstations, and the VM/CMS system.

This manual describes a step-by-step procedure for setting up the input data files and executing CometBoards to solve a structural design problem. Section 2 describes CometBoards and gives a brief theoretical background of the design and analysis methods. Instructions to prepare input data files are given in section 3. Section 4 describes the procedure for submitting a user-defined problem. Illustrative examples are given in section 5. Section 6 contains several other demonstration problems. A set of 29 problems, which have been solved by using different optimization methods, are given in appendix A. A glossary is given in appendix B, and a subject index is also provided.

Section 2

Description of CometBoards

The two key components of the code CometBoards, optimizers and analyzers, are briefly described in this section. The third component of the code, data files, is described in section 3. The central processor, which is the fourth component of the code, establishes links between the other three components to formulate and solve an optimization problem. For example, to solve an optimization problem by using any one of the seven[1] optimizers, say the sequence of unconstrained minimizations technique (SUMT), and any one of the four analyzers, say the ANALYZE/DANLYZE stiffness method (DISP), and to store the results in an output file called opt.out, give the following command to execute CometBoards:

optimize sumt disp > opt.out

CometBoards then prompts for the names of the three input files (the analysis data file, the design data file, and the optimization data file) wherein information specific to analysis, design, and optimization, respectively, should be provided for the problem. Upon solution the optimum results, consisting of the values of the objective function, the design variables, and the constraints, are stored in the output file opt.out. Further details of solving a problem by using CometBoards are given in section 4.

2.1 Design as Nonlinear Mathematical Programming Problem

CometBoards casts the minimum-weight design as a constrained nonlinear programming optimization problem utilizing input data information as follows: Find the n design variables $\vec{\chi}$ within prescribed upper and lower bounds $\left(\chi_i^L \le \chi_i \le \chi_i^U,\ i = 1, 2, \ldots, n\right)$ to make a scalar objective function $f(\vec{\chi})$, an extremum (which is to minimize the weight of the structure) subject to a set of behavior constraints formulated as a set of m inequality constraints $\left(g_j(\vec{\chi}) \ge 0,\ j = 1, 2, \ldots, m\right)$ and m_e equality constraints $\left(g_{j+m}(\vec{\chi}) = 0,\ j = 1, 2, \ldots, m_e\right)$. The behavior constraints considered are stress, displacement, and frequency limitations, and these are nonlinear explicit functions of the design variables $\vec{\chi}$. At present CometBoards' interface provides for inequality constraints only.

Stress constraints are specified as

$$g_j = \left|\sigma_j / \sigma_{j0}\right| - 1.0 \le 0 \qquad (j = 1, 2, \ldots, js) \tag{1}$$

where σ_j is the design stress in the jth element and σ_{j0} is its limiting value. Displacement constraints are specified as

$$g_{js+j} = \left|u_j / u_{j0}\right| - 1.0 \le 0 \qquad (j = 1, 2, \ldots, jd) \tag{2}$$

where u_j is the jth displacement component and u_{j0} is its limiting value. Frequency constraints are specified as

$$g_{js+jd+j} = \left(f_{j0} / f_j\right)^2 - 1.0 \le 0 \qquad (j = 1, 2, \ldots, jf) \tag{3}$$

[1]The SLP, FD, and IMSL optimizers are not available at all sites.

where f_j represents the natural frequency of the structure and f_{j0} represents the limitations on the frequency. The total numbers of stress, displacement, and frequency constraints are represented by *js*, *jd*, and *jf*, respectively.

In a mathematical programming technique the optimal design $\vec{\chi}^{\text{opt}}$ is obtained iteratively from an initial design $\vec{\chi}^0$. The design is updated at each intermediate iteration by calculating two quantities, a direction $\vec{s}_k$ and a step length α_k. The optimal design, utilizing the direction and associated step length, can be written as

$$\vec{\chi}^{\text{opt}} = \vec{\chi}^0 + \sum_{k=1}^{K} \alpha_k \vec{s}_k \tag{4}$$

The direction $\vec{s}_k$ is typically generated from the gradients of the objective function and the sensitivity of the active constraints. A one-dimensional search along direction $\vec{s}_k$ is carried out to obtain the optimum step length α_k. The updated design is then checked against specified convergence or stop criteria.

The design problem defined by equations (1) to (3) can be solved by using any one of the seven automated design schemes:

(1) Sequence of unconstrained minimizations technique (SUMT)
(2) Sequential linear programming technique (SLP)
(3) Modified method of feasible directions (FD)
(4) The IMSL version of the successive quadratic programming algorithm (IMSL)
(5) The sequential quadratic programming technique (SQP) of IDESIGN
(6) The optimality criteria methodology (OC)
(7) The fully utilized design concept (FUD)

Each of the seven optimization methods is briefly described here. Refer to specific references for further details on each method.

2.1.1 Sequence of Unconstrained Minimizations Technique (SUMT)

The sequence of unconstrained minimizations technique (SUMT), as implemented in the code NEWSUMT (ref. 1), is available through CometBoards. In the code NEWSUMT the constrained optimization problem is solved as a sequence of unconstrained minimization problems through an extended penalty function $\phi\left(\vec{\chi}, r_p\right)$, which is obtained by augmenting the constraints to the objective function and is defined as

$$\phi\left(\vec{\chi}, r_p\right) = f(\vec{\chi}) + r_p \left[\sum_{j=1}^{m} -\frac{1}{g_j(\vec{\chi})} + \sum_{i=1}^{n} \left(\frac{1}{\chi_i - \chi_i^L} + \frac{1}{\chi_i^U - \chi_i} \right) \right] \tag{5}$$

where r_p is the penalty multiplier.

In NEWSUMT the composite function $\phi\left(\vec{\chi}, r_p\right)$ defined in equation (5) has been modified to improve feasibility at intermediate designs and to accelerate one-dimensional searches. The composite function is minimized for a specified value of the penalty parameter by calculating a search direction $\vec{s}_k$ and a step length α_k and then updating the design as given in equation (4). The direction vector $\vec{s}_k$ in NEWSUMT is generalized from a modified Newton's approach $\left(\vec{s}_k = -[J]^{-1}\nabla\phi \big/ \left\|[J]^{-1}\nabla\phi\right\|\right)$. Here [J] represents the Hessian of the composite function $\phi\left(\vec{\chi}, r_p\right)$, and $\nabla\phi$ is its gradient. A golden section algorithm is followed to calculate the step length α_k. The composite function is optimized with diminishing penalty parameter r_p until convergence occurs.

The SUMT nonlinear programming technique has been successfully used in structural design optimization during the past three decades. Several other SUMT codes besides NEWSUMT are available (ref. 10).

NEWSUMT performed well for 24 examples out of a set of 29 examples, as given in appendix A. Overall, NEWSUMT appears to be a very reliable nonlinear programming code for the design optimization of structural systems. NEWSUMT can, however, be expensive, requiring extensive computations.

2.1.2 Sequential Linear Programming Technique (SLP)

A method of sequential linear programming (SLP) as implemented in the Design Optimization Tools (DOT 2.0; ref. 2) is available through CometBoards. A nonlinear design problem is converted to an SLP problem by performing a Taylor series (linear) approximation of a set of critical constraints and the objective function about a design point. Although the linear subproblem can be solved efficiently by using a standard linear programming algorithm, the implementation in DOT 2.0 uses the method of feasible directions, as described in section 2.1.3, to solve the linear subproblem. The linearization process is repeated until convergence is achieved. Convergence is achieved when the relative change in the objective function between several consecutive iterations is less than a specified tolerance.

The sequential linear programming technique performed well for approximately 9 problems out of the 29 given in appendix A. Linear programming, which moves from vertices to vertices, has the potential to avoid nonlinear design constraints (ref. 11). The SLP code is a proprietary item of Engineering Design Optimization, Inc. (ref. 2) and is not available to the general public. The SLP code, however, has been purchased for use at NASA Lewis Research Center.

2.1.3 Method of Feasible Directions (FD)

In the method of feasible directions (FD) a search direction $\vec{s}$ is determined that simultaneously satisfies two conditions: (1) the direction is feasible (i.e., it points into the feasible domain defined by a set of active constraints I_A as $\vec{s}_k^T \nabla g_j < 0, \quad j \in I_A$) and (2) the direction is usable (i.e., the objective function along the direction is reduced, $\vec{s}^T \nabla f < 0$). The direction-finding problem can be cast as a linear programming problem by using these two conditions and two parameters, which are the constraint thickness and pushoff factors. From the linear programming problem a quadratic subproblem is constructed by using dual principles to generate a direction vector. Further information on the method of feasible directions can be found in references 12 and 13. Along the search direction a minimum is obtained by polynomial approximations to the objective function and constraints.

The method of feasible directions performed well for approximately 8 of the 29 examples given in appendix A. The FD code, like SLP, is a proprietary item of Engineering Design Optimization, Inc. (ref. 2) and is also not available to the general public. The FD code, however, has been purchased for use at NASA Lewis Research Center.

2.1.4 Sequential Quadratic Programming Technique (IMSL, SQP)

Two optimization codes based on the sequential quadratic programming technique are available in the code CometBoards. The first one is the DNCONG routine of the IMSL library, which is called in CometBoards IMSL (ref. 3). The IMSL library has to be linked before running IMSL through CometBoards. The second code is IDESIGN (ref. 4). (CometBoards refers to it as SQP.)

The IMSL DNCONG routine solves the nonlinear problem as a sequence of quadratic subproblems. The subproblem is obtained by performing a quadratic approximation of the objective function and linearizing the constraints. At the *k*th intermediate design the following quadratic problem is solved:

Minimize

$$\frac{1}{2}\vec{d}^T B_k \vec{d} + \nabla f\left(\vec{\chi}_k\right)^T \vec{d}, \quad \vec{d} \in \mathbf{R}^n \tag{6}$$

subject to

$$\nabla g_i\left(\vec{\chi}_k\right)^T \vec{d} + g_i\left(\vec{\chi}_k\right) \geq 0 \quad (i = 1, \ldots, m) \tag{7}$$

$$\vec{\chi}^L - \vec{\chi}_k \le \vec{d} \le \vec{\chi}^U + \vec{\chi}_k \tag{8}$$

where

$\vec{d}$	solution of subproblem
B_k	positive definite approximation to Hessian of objective function f
$\vec{\chi}_k$	design vector at kth iteration
∇g_i	gradient of ith constraint

The step length α_k is calculated by using a simple Armijo type of bisection method combined with quadratic interpolation to decrease the augmented Lagrangian function.

The IDESIGN sequential quadratic programming method available through CometBoards is based on Pshenichny's recursive quadratic programming method. In IDESIGN the quadratic programming technique concerns the matrix $[B_k]$. The matrix $[B_k]$ approximates the Lagrangian function

$$L(\vec{\chi}, \vec{\lambda}) = f(\vec{\chi}) + \sum \lambda_j g_j(\vec{\chi}), \quad j \in I_A$$

where λ_j are the Lagrange multipliers associated with the active constraints. The step length α_k is obtained by minimizing the composite descent functions of the objective and the constraints.

On the basis of 29 examples solved by using sequential quadratic programming, both IMSL and IDESIGN (SQP) performed well, with 27 A grades for IMSL and 23 A grades for IDESIGN. Both IMSL and IDESIGN (SQP) compared well. For some problems IMSL may require fewer computations than IDESIGN (SQP); for other problems the reverse may be true (refer to appendix A).

2.1.5 Optimality Criteria Methods (OC)

The optimality criteria (OC) design code available through CometBoards has been developed at NASA Lewis Research Center (ref. 5). The OC code incorporates OC methods available in the literature (ref. 14) with their generalization for stress constraints. The OC methods can be considered as variants of the Lagrange multiplier approach that have been specialized to structural design problems (refs. 14 and 15). The Lagrangian L in OC methods is formed by adjoining behavior constraints to the objective function through Lagrange multipliers, and it is defined as

$$L(\vec{\chi}, \vec{\lambda}) = f(\vec{\chi}) + \sum_{j}^{I_A} \lambda_j g_j(\vec{\chi}) \tag{9}$$

where λ_j are the Lagrange multipliers and the summation is taken over the set of active constraints I_A. The stationary conditions of the Lagrangian L with respect to the design variables $\vec{\chi}$ and the Lagrange multipliers $\vec{\lambda}$ yield two sets of equations. Simultaneous solution of these two sets of equations should yield the solution to the optimization problem. The optimality criteria method follows an iterative scheme. Lagrange multipliers are updated first. The multipliers are then used to update the design variables. Several optimality criteria update formulas are available (refs. 14 and 15). The exponential form of the Lagrange multiplier update formula is

$$\lambda_{j,k+1} = \lambda_{j,k}\left(1 + g_j\right)^{\gamma^k p_0} \tag{10}$$

where

$\lambda_{j,k}$	value of Lagrange multiplier associated with *j*th active constraint at *k*th iteration
p_0	initial value of update parameter
γ	acceleration parameter used to modify update parameter

The linearized form of the design variable update formula is

$$\chi_{i,k+1} = \chi_{i,k}\left\{ 1 - \frac{\left[1 + \left(\sum_{j}^{I_A} \lambda_j \nabla g_{i,j} \right) \Big/ \nabla f_i \right]}{\hat{\gamma}^k q_0} \right\} \tag{11}$$

where

$\chi_{i,k}$	value of *i*th component of design variable at *k*th iteration
$\nabla g_{i,j}$	(*i*,*j*)th component of gradient matrix of active constraints
∇f_i	*i*th component of gradient of objective function
$q_0, \hat{\gamma}$	acceleration parameters

The design obtained is then scaled to ensure constraint feasibility. The iteration process continues until convergence is achieved or the maximum number of iterations is exceeded.

Several hybrid techniques have been included under optimality criteria methods. A hybrid technique is an attempt to improve the performance of the optimality criteria method by combining the good features of the OC with the good features of the fully stressed design concept. A hybrid technique generates a first design for displacement and frequency constraints by using the OC method; next it calculates a second design for stress constraints by using the fully stressed design concept. Both designs are melanged to obtain a design that satisfies stress, displacement, and frequency constraints. The hybrid formulation is explained in more detail in reference 5.

The performance of the OC method, on the basis of examples solved (ref. 5), was found to be satisfactory for problems with few active constraints or with a small number of design variables. For problems with large numbers of behavior constraints and design variables, the OC method appears to follow a subset of active constraints that can result in a heavier design. Of the 29 examples given in appendix A the OC method achieved grade A for 13 problems.

2.1.6 Fully Utilized Design Technique (FUD)

The fully utilized design (FUD) code available in CometBoards has been developed at NASA Lewis Research Center (ref. 6). The stress-ratio-based fully stressed design concept, which is applicable for stress constraints, has been extended to include displacement and frequency constraints. Fully utilized design is obtained by scaling the fully stressed design to satisfy the maximum violated displacement and frequency constraints $\left[\vec{\chi}^{\text{opt}} = \vec{\chi}^{\sigma,\text{opt}}\left(1 + g_{\max}\right)\right]$, where $g_{\max}$ is the value of the largest violated displacement or frequency constraint and $\vec{\chi}^{\sigma,\text{opt}}$ is the fully stressed design.

In the 29 examples solved by using the fully utilized design technique, FUD provided an acceptable design in a very short computation time only when stress constraints dominated the design. FUD scored 8 A grades out of 29, as shown in appendix A.

2.2 Analysis Methods

Modest finite element analysis methods based on the displacement and integrated force methods have been incorporated in CometBoards to validate and check out the computer code and to ascertain the relative performance of the different design methods. The analysis codes available in CometBoards are

(1) Displacement method (DISP)
(2) Integrated force method (IFM)
(3) Three versions of simplified integrated force method (SIFMSD, SIFMD, and SIFMS)
(4) Closed-form solution for a three-bar truss (CFA)

The codes perform linear elastic analysis of structures with small displacements, providing static analysis under concentrated loads and free vibration analysis. Each analyzer determines the response parameters (stresses, strains, displacements, frequencies, etc.) for a given structure and given load conditions from information provided in the finite element analysis data file. The response parameters are used to formulate the behavior constraints on stresses, displacements, and frequencies as given by equations (1) to (3). The finite element information, such as elements, nodal coordinates, and thickness, is used to formulate weight as the objective function. The analyzer generates the gradients of the constraints in closed form by using matrix differentiation and chain rules. CometBoards also provides an option for determining the gradients of the constraints and objective functions numerically by using the finite difference scheme. The analysis methods are briefly described here.

2.2.1 Displacement Method

The displacement method of analysis available in CometBoards is called DISP for code execution. This analysis computer code was developed by the Air Force in the late 1970's, and it is popularly known as ANALYZE/DANLYZE—ANALYZE representing analysis for multiple static load conditions and DANLYZE standing for dynamic analysis (ref. 7). The ANALYZE/DANLYZE code is a modest finite element analyzer that includes both stress analysis and sensitivity analysis of a structure which can be modeled by four types of elements with arbitrary orientation in three-dimensional space. The elements are

(1) A two-node truss element, with element type identified as "2" in the keyword card CONNECTIVITY: (see page 14)
(2) A three-node membrane triangular element, with element type identified as "3" in the keyword card CONNECTIVITY:
(3) A four-node quadrilateral membrane, with element type identified as "4" in the keyword card CONNECTIVITY:
(4) A four-node shear panel, with element type identified as "5" in the keyword card CONNECTIVITY:

The ANALYZE/DANLYZE analysis code can be used to specify constraints on stresses (a multiaxial stress field constraint is imposed on von Mises stress), nodal displacements, and frequencies. Only one stress constraint at the center of gravity of the element is imposed for each element. The material properties are assumed to be isotropic.

2.2.2 Integrated Force Method

The integrated force method available in CometBoards is called IFM for code execution. This method is a finite-element-based code wherein forces are treated as the primary variables. IFM also replaces the continuum with a discrete model, but IFM explicitly constrains the primary force variables {F} to satisfy both the equilibrium equations and the compatibility conditions.

The basic IFM equations for static analysis are presented here. Refer to reference 17 for more information on IFM. A discrete or discretized structure for analysis has *n* force and *m* displacement degrees of freedom. It has *m* equilibrium equations and $r = (n - m)$ compatibility conditions. The *m* equilibrium equations, $[B]\vec{F} = \vec{P}$, and the *r* compatibility conditions, $[C][G]\vec{F} = \delta\vec{R}$, are coupled to obtain the governing equations of the IFM as

$$\left[\frac{[B]}{[C][G]}\right]\vec{F} = \left[\frac{\vec{P}}{\delta\vec{R}}\right] \qquad \text{or} \qquad [S]\vec{F} = \vec{P}^{*} \tag{12}$$

where

[B]	$m \times n$ equilibrium matrix
[C]	$r \times n$ compatibility matrix
[G]	$n \times n$ concatenated flexibility matrix
$\vec{P}$	*m*-component load vector
$\delta\vec{R}$	*r*-component effective initial deformation vector
[S]	$n \times n$ IFM governing matrix

and $\delta\vec{R} = -[C]\vec{\beta}_0$, where $\vec{\beta}_0$ is the *n*-component initial deformation vector (taken to be zero). The matrices [B], [C], [G], and [S] are banded, and they have full-row ranks of *m*, *r*, *n*, and *n*, respectively. The solution of equation (12) yields the *n* forces $\vec{F}$. The *m* displacements $\vec{X}$ are obtained from the forces $\vec{F}$ by backsubstitution:

$$\vec{X} = [J][G]\vec{F} + \vec{\beta}_0 \tag{13}$$

where [J] is the $m \times n$ deformation coefficient matrix defined as

$$[J] = m \text{ rows of } [S]^{-T} \tag{14}$$

Equations (12) and (13) represent the two key IFM relations for finite element analysis. Two types of constraints can be specified with this analyzer: stress and displacement. The only finite element model available in the IFM analyzer is a bar (or truss) element. Isotropic material properties are assumed. This analyzer is not recommended when more than 60 elements or more than 200 constraints are present.

2.2.3 Simplified Integrated Force Method

The simplified integrated force method is identical to the integrated force method except for some simplifications while calculating the sensitivities of stress and displacement constraints. Three versions of the simplified integrated force method are available in CometBoards:

(1) SIFMSD: Both stress and displacement sensitivities are simplified.
(2) SIFMD: Displacement sensitivities only are simplified.
(3) SIFMS: Stress sensitivities only are simplified.

For execution of CometBoards these analyzers are called SIFMSD, SIFMD, and SIFMS.

2.2.4 Closed-Form Analysis Method

The closed-form analysis (CFA) of a specific three-bar truss problem, available in CometBoards, is illustrated in section 6. CFA is an analytical solution based on integrated force method analysis. Stress, displacement, and frequency constraints can be specified with this method.

The simple closed-form analyzer is very useful for checking out the other three analyzers for constraints and sensitivities as well as for validating all the optimization methods. A new user of the design code is urged to use this simple example to begin working with CometBoards. For code execution this analyzer is called CFA.

2.3 Description of Interface Module

The interface module consists of two parts: a C-shell script (or REXX executive, on VM/CMS) and a set of Fortran routines. The shell script (or executive) interprets the command line entered by the user. It then loads appropriate modules based on the optimizer and analyzer chosen. In addition, the shell script (or executive) determines whether the user has chosen to run a demonstration problem or a problem defined by the user. For a demonstration problem it links three input data files for the demo chosen to the loaded program. Otherwise, the script (or executive) links three user-specified input data files, or it prompts the user for the file names if valid names were not included on the command line. Then the script (or executive) links program output units to output files as named on the command line. Finally, execution of the program is invoked and, after completion, scratch files are removed.

The second part of the interface module consists of the main program and associated subroutines. It reads the problem specification, such as the geometry, material properties, and design limitations of the finite element model from keyword-based, free-format data files. It also reads a data file containing only modifications to the default values for the optimization parameters. Interactive input of some of the optimization parameters is an additional optional feature with the OC optimizer. This part of the interface module initializes variables, calls the optimizer, and outputs the final results.

Section 3
CometBoards Data Management

CometBoards has a total of 10 files for the management of input and output data. Only a few of these 10 files are required to solve an optimization problem. Other files are provided for intermediate printing from analyzers and optimizers during the execution of CometBoards. The three principal data files required to run cometboards are (1) anldat, an acronym for analysis data; (2) dsgndat, for design data; and (3) optdat, for optimization data. The optimality criteria method has an additional optional feature (i.e., to read some of the optimization data directly from the screen on an interactive basis). The standard output file contains the optimization results, including the optimum values of the objective function, the design variables, and the constraints.

The three input files (anldat, dsgndat, and optdat) read data by using keywords followed by their data. Keywords with their data can be entered in an arbitrary sequence with a few exceptions that are described in subsequent sections. Each keyword should appear only once in a data file. The CometBoards input processor requires full keywords or their alias form, including any blanks, in mixed case exactly as specified. The input data file processor checks for possible typing errors and data inconsistency. When a mistake is detected, the program prints an error message, reads the remaining input data, and then terminates execution. Errors handled by the Fortran input/output routines will cause immediate program termination, however.

A keyword is followed by an exclamation point (!), an equal sign (=), or a colon (:) on the same line. A keyword followed by an exclamation point indicates that no value is to be read after this keyword. A keyword followed by an equal sign indicates that a numeric value must be supplied on the same line following the equal sign. If the keyword begins with a letter between I and N, inclusive, the numeric value must be an integer; otherwise the value should be real. A keyword followed by a colon indicates that a list of values are to be read on the line (or lines) immediately following the keyword line. In this case the numeric values must be entered as integers and/or reals, as applicable for the keyword. When real values are encountered where integers are expected or characters are encountered where numeric values are expected, Fortran input/output routines will generate error messages and the program will abnormally terminate.

Comment lines may be included to explain input data. Comment lines begin with a dollar ($) sign. Comments may only appear immediately prior to a keyword line. Comments may not appear within a set of data lines. Blank lines are ignored. Numbers on data lines can be separated by blanks or commas. Whenever a large number of numeric values are required on a data line, those values may be continued onto subsequent lines. Nonetheless, each data line must begin on a new line.

The 10 file units used in CometBoards are listed in table 1. Outputs contained in file units 8, 10, and 11 are retained only at the request of the user (see section 4). File unit 66 is not normally saved.

TABLE 1.—CometBoards INPUT/OUTPUT FILES

File unit	Contents
2 (input)	anldat (analysis input data)
3 (input)	dsgndat (design input data)
4 (input)	optdat (optimization input data)
5 (input)	Interactive input
6 (output)	Echo of input, error messages, iteration data, and output results
7 (output)	Interactive prompts
8 (output)	Weight-versus-reanalysis-cycles output
10 (output)	Weight-versus-CPU-time output
11 (output)	Output of design, objective, constraints, and gradients for each analysis cycle
66 (output)	Dummy file used for debugging

3.1 Analysis Data File, ANLDAT

The analysis data file, ANLDAT, reads the data required for finite element analysis of the structure, such as nodal coordinates, element connectivity, material properties, external loads, and lumped mass distribution. ANLDAT uses 26 keywords to read analysis data. The keywords are listed in table 2, in the order in which they are typically entered in ANLDAT. The keywords are described, however, in alphabetical order by their short names. Each keyword (short name or long name) may be entered in mixed, upper, or lower case letters. Certain keywords are mandatory and others are optional. The optional keywords have default values. Some keywords must be given in a specified order in the data file. The 26 keywords that may be encountered in ANLDAT, in alphabetical order, are explained on pages 14 to 40. Note that due to restrictions within the displacement analyzer, special consideration may need to be given during layout of structures containing membrane or shear panel elements in regard to the numbering of the nodes. See keyword CONNECTIVITY: for details.

TABLE 2.—TYPICAL KEYWORD SEQUENCE IN ANLDAT

Short name	Long name	Default	Requirement
TITLE:	TITLE:	All blanks	Optional
IDIM=	I DIMENSION=	3	↓
CORDUIN!	COORDINATES IN INCHES!	inches	
CORDUFT!	COORDINATES IN FEET!	inches	
LDUKIPS!	LOADS IN KIPS!	kips	
LDULBS!	LOADS IN POUNDS!	kips	
MODUKSI!	YOUNGS MODULUS IN KSI!	ksi	
MODUPSI!	YOUNGS MODULUS IN PSI!	ksi	
DENUPCI!	DENSITY IN LBS/CUBIC IN!	(Only option)	
MATPROP:	MATERIAL PROPERTIES:	(Steel)	↓
CONN:	CONNECTIVITY:	------------------	Mandatory
COORD:	COORDINATES:	------------------	↓
NREST:	NODAL RESTRAINTS:	------------------	
LCON:	LOAD CONDITIONS:	------------------	↓
IPRINT=	I PRINT CONTROL=	0	Optional
DYNBEG!	BEGIN DYNAMIC ANALYSIS PARMS!	(None)	↓
FREQPARM=	FREQ ANALYSIS CONV PARM =	0.02	
FREQSWP:	FREQUENCY SWEEP:	(See DSGNDAT)	
FREQDISC:	DISCRETE FREQUENCIES:	(See DSGNDAT)	
LMUKIPS!	MASS IN KIPS-F!	kips-f	
LMULBS!	MASS IN POUNDS-F!	kips-f	
NOMASS!	NO ELEMENTAL MASS!	(None)	↓
SMASS!	CONSISTENT ELEMENTAL MASS!	↓	(a)
LMASS!	LUMPED ELEMENTAL MASS!		(a)
EMASS:	EQUIPMENT LUMPED MASS:		(a)
DYNEND!	END DYNAMIC ANALYSIS PARMS!	↓	(b)

[a]At least one of the three is mandatory for dynamic analysis.
[b]Mandatory for dynamic analysis.

The following four keywords are mandatory:

CONNECTIVITY:
COORDINATES:
NODAL RESTRAINTS:
LOAD CONDITIONS:

In addition, when the keyword BEGIN DYNAMIC ANALYSIS PARMS! is present, one of the following keywords is also mandatory:

EQUIPMENT LUMPED MASS:
CONSISTENT ELEMENTAL MASS!
LUMPED ELEMENTAL MASS!

The keyword BEGIN DYNAMIC ANALYSIS PARMS! should be concluded with the keyword END DYNAMIC ANALYSIS PARMS!, such as given on pages 20 and 21.

The following two keywords must be entered in a specific order:

I DIMENSION= The dimension of the problem must be specified before NODAL RESTRAINTS:, LOAD CONDITIONS:, and COORDINATES: are specified.

MATERIAL PROPERTIES: If more than one material property is given, this keyword must be specified before CONNECTIVITY:.

Default values specified for optional keywords in the analysis data file are

TITLE: (all blanks)

I DIMENSION = 3
COORDINATES IN INCHES!
LOADS IN KIPS!
YOUNGS MODULUS IN KSI!
DENSITY IN LBS/CUBIC IN! (only option)
MATERIAL PROPERTIES: 30000.0 0.3 0.284 (steel)
I PRINT CONTROL = 0
(no dynamic analysis; i.e, no BEGIN DYNAMIC ANALYSIS PARMS! keyword)
FREQ ANALYSIS CONV PARM = 0.02
DISCRETE FREQUENCIES: (as specified in the DSGNDAT data file)
MASS IN KIPS-F!
NO ELEMENTAL MASS!

CONNECTIVITY: **CONN:**

Description: This keyword is used to specify the element type identifier and associated node numbers for each element in the finite element model. The element type identifiers are

Element type identifier	Element type	Number of nodes, *N*
2	Bar or truss	2
3	Triangular membrane	3
4	Quadrilateral membrane	4
5	Shear panel	4

Subsequent data lines: Contain integer numbers as follows:
If the number of material properties is one, enter, one line per element,

Integer 1: Element type identifier
Integers 2 to N+1: Associated node numbers for element, in correct order (see Remarks)

Data for this keyword are terminated when a zero integer value is encountered on the last line.

If the number of material properties is greater than one, enter, one line per element,

Integer 1: Material property number
Integer 2: Element type identifier
Integers 3 to N+2: Associated node numbers for element, in correct order (see Remarks)

Data for this case are terminated when two zero integer values are encountered on the last line.

Note: The maximum number of elements allowed is 490.

Example 1: **CONNECTIVITY:**

```
2  1  2
2  1  3
2  1  4
0
```

(Here one material property is specified for a three-bar truss whose elements are connected to four nodes; see figure 3 on page 124.)

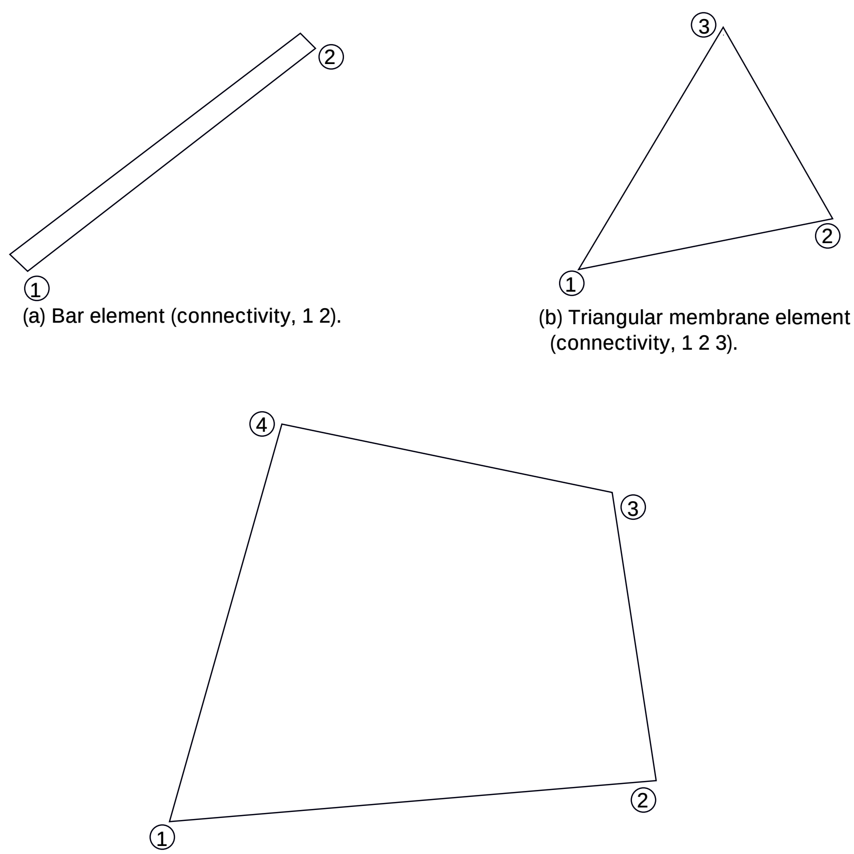
Example 2: **CONNECTIVITY:**

1 2 1 2
1 2 1 3
2 2 1 4
0 0

(Here two material properties are specified: The first material property is specified for the first two elements of the three-bar truss, and the second is specified for the third element of the truss; see figure 3.)

Remarks: This keyword is required.

For bar and triangular membrane elements the nodal connectivity should be specified from lowest to highest node numbers, as shown in figure 2. For quadrilateral membrane and shear panel elements the first node specified must be the lowest node number, and the second node specified must be the second lowest number. Moreover, the node numbers must be listed in the order in which they appear on the element (in a counterclockwise fashion). Note that these last two restrictions, which are necessary for correct frequency analysis by the displacement analyzer, will require some thought during the element and node numbering of the structure.



(a) Bar element (connectivity, 1 2).

(b) Triangular membrane element (connectivity, 1 2 3).

(c) Quadrilateral or shear panel (connectivity, 1 2 3 4).

Figure 2.— Elements and connectivity.

COORDINATES: **COORD:**

Description: This keyword is used to specify the nodal coordinates in the Cartesian coordinate system (*X*, *Y*, and *Z*), where *Z* is included only if the dimension of the problem is three.

Subsequent data lines: Contain *N* real numbers per line, where *N* is the dimension of the problem specified (i.e., 2 or 3). To terminate the coordinate input, enter *N* real parameters that are negative and have a very large magnitude (e.g., -1×10^{75}).

Reals 1 to N: Coordinates of node

Note: The maximum number of nodes allowed is 165.

Example: **COORDINATES:**

0.000	**0.000**
–100.000	**100.000**
0.000	**100.000**
100.000	**100.000**
–1.E75	**–1.E75**

(Here the *X* and *Y* coordinates of a two-dimensional, three-bar-truss problem are specified; see figure 3.)

Remarks: This keyword is required.

The default unit of measure for this keyword is inches.

COORDINATES IN FEET! **CORDUFT!**

Description: This optional keyword is used to specify that nodal coordinates are given in feet.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Coordinates are in inches.

Example: **COORDINATES IN FEET!**

COORDINATES IN INCHES! **CORDUIN!**

Description: This optional keyword is used to specify that nodal coordinates are given in inches.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Coordinates are in inches.

Example: **COORDINATES IN INCHES!**

DENSITY IN LBS/CUBIC IN! **DENUPCI!**

Description: This optional keyword is used to specify that density is given in pounds force per cubic inch.

Data value: No data follow the keyword (just enter the keyword alone).

Default: Density is in pounds per cubic inch.

Example: **DENSITY IN LBS/CUBIC IN!**

Remarks: This is the ONLY option.

BEGIN DYNAMIC ANALYSIS PARMS! **DYNBEG!**

Description: This optional keyword is used to indicate that dynamic analysis is to be performed by the analyzer.

Data value: No data follow this keyword (just enter the keyword alone).

Default: No dynamic analysis

Example: **BEGIN DYNAMIC ANALYSIS PARMS!**

Remarks: This keyword should be given one time only. At the end of dynamic analysis input the keyword END DYNAMIC ANALYSIS PARMS! should be given.

END DYNAMIC ANALYSIS PARMS! **DYNEND!**

Description: This keyword is used to terminate all the input for dynamic analysis.

Data value: No data follow this keyword (just enter the keyword alone).

Example: **END DYNAMIC ANALYSIS PARMS!**

Remarks: This keyword is required if BEGIN DYNAMIC ANALYSIS PARMS! was given.

EQUIPMENT LUMPED MASS: **EMASS:**

Description: This optional keyword is used to specify the equipment lumped masses.

Subsequent data lines: Contain an integer number followed by a real number, one per line. Enter a pair of zeros (one integer and one real) to end the equipment mass input.

Integer 1: Node number
Real 1: Value of mass

To terminate this input, enter two zeros (one integer and one real) on the last line.

Default: No equipment masses

Note: The maximum number of equipment lumped masses allowed is 162.

Example: **EQUIPMENT LUMPED MASS:**

1 262.5
0 0.0

(Here one lumped mass of 262.5 kips-force is specified at node 1.)

Remarks: The value of each mass specified must be positive. When dynamic analysis is present, one of the following keywords must be included:

CONSISTENT ELEMENTAL MASS!
LUMPED ELEMENTAL MASS!
EQUIPMENT LUMPED MASS:

The default unit of measure for this keyword is kips-force.

DISCRETE FREQUENCIES: **FREQDISC:**

Description: This optional keyword is used to specify the mode numbers of the natural frequencies of the structure that are to be calculated.

Subsequent data lines: Contain one integer number per line.

Integer 1: Mode number

To terminate this input, enter a zero (integer) on the last line.

Default: Only those frequency modes that are used as constraints (as specified in the design data file) are calculated.

Note: The maximum number of frequency modes that can be calculated is five.

Example: **DISCRETE FREQUENCIES:**

```
 1
 5
10
 0
```

(Here the calculation of three frequency modes is requested.)

FREQ ANALYSIS CONV PARM= **FREQPARM=**

Description: This optional keyword is used to specify the percentage of error allowed in frequency calculations.

Data value: Following the keyword, enter one real number.

Real 1: Frequency calculation percentage of error allowed

Default: The frequency calculation percentage of error is 0.02.

Example: **FREQ ANALYSIS CONV PARM = 0.001**

Remarks: This parameter must be positive.

FREQUENCY SWEEP: **FREQSWP:**

Description: This optional keyword is used to specify a range of mode numbers for the natural frequencies of the structure that are to be calculated.

Subsequent data line: Contains one pair of integers.

Integer 1: First mode number of range desired
Integer 2: Last mode number of range desired

Default: Only those frequency modes that are used as constraints (as specified in the design data file) are calculated.

Note: The maximum number of frequency modes that can be calculated is five.

Example: **FREQUENCY SWEEP:**

2 4

(Here calculation of the second, third, and fourth frequency modes is requested.)

Remarks: Note that the first and last modes specified are calculated, along with all modes in between.

I DIMENSION= **IDIM=**

Description: This optional keyword is used to specify the dimension of the problem.

Data value: Following the keyword, enter one integer number specifying the dimension of the problem.

Integer 1: 2 For two-dimensional problem
3 For three-dimensional problem

Default: I DIMENSION = 3

Example: **I DIMENSION = 2**

Remarks: This keyword, when present, must precede the following keywords:

NODAL RESTRAINTS:
LOAD CONDITIONS:
COORDINATES:

I PRINT CONTROL= **IPRINT=**

Description: This optional keyword is used to specify the level of print output.

Data value: Following the keyword, enter one integer number ranging from zero to 999 (see Remarks).

Integer 1: Number ranging from zero to 999

Default: The print control is zero.

Example: **I PRINT CONTROL = 2**

(In this case, weight, energy, and displacements are printed if the displacement method analyzer is used; see Remarks.)

Remarks: This keyword only affects printed output from the displacement method analyzer. Three types of printed output are controlled by this keyword: (1) static analysis, which is controlled by the units place in the parameter specified; (2) global matrices, which is controlled by the tens place; and (3) eigenanalysis, which is controlled by the hundreds place. The weight and energy are printed if the units place is nonzero. Stresses are printed if the units place is 1, displacements if it is 2, and both stresses and displacements if it is 3. For the tens and hundreds places the larger the digit (up to 3), the more output is printed.

LOAD CONDITIONS: **LCON:**

Description: This keyword is used to specify the load components associated with each load condition.

Subsequent data lines: Contain integer and real numbers as shown below. The directions are indicated by the integers 1, 2, and 3, corresponding to the *X*, *Y*, and *Z* directions, respectively. Enter two zeros (integers) in one line to end each load condition. Enter an extra line with two zeros (integers) to terminate all the load input.

Integer 1: Node number
Integer 2: Number of load components *N* at that node
Integer 3: Direction of load
Real 1: Value of load

Integer 3 and real 1 are repeated *N* times. To end each load condition, enter two zeros (integers). To terminate all the load input, enter two lines of two zeros per line (integers).

Note: The maximum number of load conditions allowed is five.

Example: **LOAD CONDITIONS:**

1 2 1 50.00 2 100.00
0 0
0 0

(Here one load condition is specified with two load components at node 1: 50 kips in the *X* direction and 100 kips in the *Y* direction.)

Remarks: This keyword is required.

The default unit of measure for this keyword is kips.

LOADS IN KIPS! **LDUKIPS!**

Description: This optional keyword is used to specify that loads are given in kips.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Loads are in kips.

Example: **LOADS IN KIPS!**

LOADS IN POUNDS! **LDULBS!**

Description: This optional keyword is used to specify that loads are given in pounds.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Loads are in kips.

Example: **LOADS IN POUNDS!**

LUMPED ELEMENTAL MASS! **LMASS!**

Description: This optional keyword is used to indicate that lumped elemental mass formulation will be used.

Data value: No data follow this keyword (just enter the keyword alone).

Default: No elemental mass

Example: **LUMPED ELEMENTAL MASS!**

Remarks: When dynamic analysis is present, one of the following keywords must be included:

CONSISTENT ELEMENTAL MASS!
LUMPED ELEMENTAL MASS!
EQUIPMENT LUMPED MASS:

This keyword should not be used when either one of these two keywords is present:

CONSISTENT ELEMENTAL MASS!
NO ELEMENTAL MASS!

MASS IN KIPS-F! **LMUKIPS!**

Description: This optional keyword is used to specify that equipment lumped masses are given in kips-force.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Mass is in kips-force.

Example: **MASS IN KIPS-F!**

MASS IN POUNDS-F! **LMULBS!**

Description: This optional keyword is used to specify that equipment lumped masses are given in pounds-force.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Mass is in kips-force.

Example: **MASS IN POUNDS-F!**

MATERIAL PROPERTIES: **MATPROP:**

Description: This optional keyword is used to define sets of material properties: Young's modulus, Poisson's ratio, and density. This keyword, if present, must precede the keyword CONNECTIVITY:.

Subsequent data lines: Contain three real parameters for each set of material properties, one set per line. Enter three zeros (reals) when done.

Real 1: Young's modulus
Real 2: Poisson's ratio
Real 3: Density

The material properties keyword is terminated with three real zeros on the last line.

Default: 30000.000 0.300 0.284 (steel for all elements)

Note: The maximum number of sets of material properties allowed is four.

Example: **MATERIAL PROPERTIES:**

30000.000 0.300 0.284
10000.000 0.300 0.100
0.0 0.0 0.0

(Here two sets of material properties are specified: one for steel and another for aluminum.)

Remarks: Each Young's modulus and density specified must be positive. Each Poisson's ratio must be in the interval 0.0 to 1.0.

The default units of measure for this keyword are Young's modulus in kips per square inch and density in pounds per cubic inch.

YOUNGS MODULUS IN KSI! **MODUKSI!**

Description: This optional keyword is used to specify that Young's modulus is given in kips per square inch.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Young's modulus is in kips per square inch.

Example: **YOUNGS MODULUS IN KSI!**

YOUNGS MODULUS IN PSI! **MODUPSI!**

Description: This optional keyword is used to specify that Young's modulus is given in pounds per square inch.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Young's modulus is in kips per square inch.

Example: **YOUNGS MODULUS IN PSI!**

NO ELEMENTAL MASS! **NOMASS!**

Description: This optional keyword is used to indicate that the elements will be assumed to be massless.

Data value: No data follow this keyword (just enter the keyword alone).

Default: No elemental mass

Example: **NO ELEMENTAL MASS!**

Remarks: When this keyword and dynamic analysis are present, the following keyword is also required:

EQUIPMENT LUMPED MASS:

This keyword should not be used when either one of these two keywords is present:

CONSISTENT ELEMENTAL MASS!
LUMPED ELEMENTAL MASS!

Rather than setting the elemental mass identically to zero, it is set to a small value in order to prevent singularities in the mass matrix.

NODAL RESTRAINTS: **NREST:**

Description: This keyword is used to specify the degrees of freedom of those nodes that are restrained.

Subsequent data lines: Contain integer numbers, one set of integers per line, as specified below. (1 = *X* direction, 2 = *Y* direction, and 3 = *Z* direction). Enter two zeros (integers) to end this input.

Integer 1: Restrained node number
Integer 2: Number of degrees of freedom *N* associated with restrained node number
Integers 3 to N+2: Direction in which each degree of freedom is restrained

To terminate this input, enter two zeros (integers) on the last line.

Note: The maximum number of restrained degrees of freedom allowed is 100.

Example: **NODAL RESTRAINTS:**

2 2 1 2
3 2 1 2
4 2 1 2
0 0

(Here nodes 2, 3, and 4 are restricted in both the *X* and *Y* directions.)

Remarks: This keyword is required.

CONSISTENT ELEMENTAL MASS! **SMASS!**

Description: This optional keyword is used to indicate that consistent elemental mass formulation will be used. (This option is not available when quadrilateral elements are present.)

Data value: No data follow this keyword (just enter the keyword alone).

Default: No elemental mass

Example: **CONSISTENT ELEMENTAL MASS!**

Remarks: When dynamic analysis is present, one of the following keywords must be included:

CONSISTENT ELEMENTAL MASS!
LUMPED ELEMENTAL MASS!
EQUIPMENT LUMPED MASS:

This keyword should not be used when either one of these two keywords is present:

LUMPED ELEMENTAL MASS!
NO ELEMENTAL MASS!

 TITLE:

Description:	This optional keyword contains an alphanumeric description of the problem to be solved.
Subsequent data line:	Contains alphanumeric characters.
Default:	All blanks
Example:	**TITLE:** **Three bar truss.**
Remarks:	The maximum length is 80 characters.

3.2 Design Data File, DSGNDAT

The design data file, DSGNDAT, reads data required to formulate the design as a nonlinear programming problem (see section 2.1), such as stress, displacement, and frequency limitations, bounds on design variables, initial design, and design variable linking.

One tool that has been introduced for improving convergence of optimization methods by reducing the number of design variables in a given problem is the linking of design variables into linked groups. Specifically, the original design variables for a given problem are divided into a small number of groups. Within each group one of the original design variables is chosen to represent that group as the "linked design variable." Linking factors are then assigned to all other elements of the group, whereby the area of each element is related to the chosen "representative" element by the fixed linking factor.

The DSGNDAT file uses 25 keywords to read the design data as listed in table 3 in the order in which they are typically entered. The keywords are described, however, on the following pages in alphabetical order by their short names. As in ANLDAT, keywords may be entered in upper, lower, or mixed case. Here all keywords are optional because default values have been incorporated in CometBoards. Certain keywords, when present, must be given in a specified order, as described below. The restrictions and default values for the DSGNDAT keywords are as follows.

Only one method each is allowed for specifying lower bounds, upper bounds, initial design, and linking factors:

(1) To specify lower bounds, only one of the following four keywords is allowed:

DISCRETE DV LOWER BOUNDS:
UNIFORM DV LOWER BOUNDS=

TABLE 3.—TYPICAL KEYWORD SEQUENCE IN DSGNDAT

Short name	Long name	Default
STRUKSI!	STRESS IN KSI!	ksi
STRUPSI!	STRESS IN PSI!	ksi
STRENGTH:	STRESS LIMITS:	1.0×10^{10}
DISPUIN!	DISPLACEMENTS IN INCHES!	inches
DISPUFT!	DISPLACEMENTS IN FEET!	inches
STIFFN:	DISP LIMITS BY NODES:	(No displacement limits)
STIFFD:	DISP LIMITS BY DIRECTION:	
FREQL:	FREQUENCY LIMITS:	(No frequency limits)
LINKDV:	DESIGN VARIABLE LINKING:	(No linking)
UNITLFACT!	LINKING FACTORS OF UNITY!	Unity
ANLLFACT!	CALCULATE LINKING FACTORS!	------------------------------
GIVLFACT:	LINKING FACTORS FOLLOW:	------------------------------
DVLBU=	UNIFORM DV LOWER BOUNDS=	0.01
DVUBU=	UNIFORM DV UPPER BOUNDS=	1.0×10^{10}
LDVLBU=	UNIFORM LINKED DV LOW BOUNDS=	0.01
LDVUBU=	UNIFORM LINKED DV UP BOUNDS=	1.0×10^{10}
DVLBD:	DISCRETE DV LOWER BOUNDS:	------------------------------
DVUBD:	DISCRETE DV UPPER BOUNDS:	------------------------------
LDVLB:	LINKED DV LOWER BOUNDS:	------------------------------
LDVUB:	LINKED DV UPPER BOUNDS:	------------------------------
VBMAX=	VARIABLE BOUNDS FACTOR=	10 000.0
UNITID!	INITIAL DESIGN OF UNITY!	Unity
ANLID!	CALCULATE INITIAL DESIGN!	------------------------------
GIVID:	INITIAL DESIGN FOLLOWS:	------------------------------
GIVLID:	LINKED INITIAL DESIGN:	------------------------------

LINKED DV LOWER BOUNDS:
UNIFORM LINKED DV LOW BOUNDS=

(2) To specify upper bounds, only one of the following four keywords is allowed:

DISCRETE DV UPPER BOUNDS:
UNIFORM DV UPPER BOUNDS=
LINKED DV UPPER BOUNDS:
UNIFORM LINKED DV UP BOUNDS=

(3) To specify initial design, only one of the following four keywords is allowed (see also the following paragraph):

INITIAL DESIGN OF UNITY!
LINKED INITIAL DESIGN:
INITIAL DESIGN FOLLOWS:
CALCULATE INITIAL DESIGN! (not yet implemented)

(4) To specify linking factors, only one of the following three keywords is allowed (see also the following paragraph):

LINKING FACTORS OF UNITY!
LINKING FACTORS FOLLOW:
CALCULATE LINKING FACTORS!

When linking of design variables is specified:

(1) The keywords for linking design variables must be given in the following order:

DESIGN VARIABLE LINKING: This keyword must be specified before any of the following keywords:

LINKED INITIAL DESIGN:
LINKED DV LOWER BOUNDS:
LINKED DV UPPER BOUNDS:
CALCULATE LINKING FACTORS!

(2) The valid combinations for determining linking factors and initial design are given in table 4. For METHLF the methods of specifying the linking factors are

TABLE 4.—COMBINATIONS FOR LINKING FACTORS (METHLF) AND INITIAL DESIGN METHODS (METHID)

METHID	METHLF		
	1	2	3
1	Yes	Yes	No
2	Yes	Yes	No
3	No	No	Yes
4	No	No	Yes

METHLF = 1 (LINKING FACTORS OF UNITY!, default)
METHLF = 2 (LINKING FACTORS FOLLOW:)
METHLF = 3 (CALCULATE LINKING FACTORS!)

and for METHID the methods of specifying the initial design are

METHID = 1 (INITIAL DESIGN OF UNITY!, default)
METHID = 2 (LINKED INITIAL DESIGN:)
METHID = 3 (INITIAL DESIGN FOLLOWS:)
METHID = 4 (CALCULATE INITIAL DESIGN!, not yet implemented)

Moreover, if METHLF = 3, CALCULATE LINKING FACTORS! must follow either INITIAL DESIGN FOLLOWS: or CALCULATE INITIAL DESIGN!

The default values in the design data file are

STRESS IN KSI!
STRESS LIMITS: 25.0 25.0 16.0 (steel)
INITIAL DESIGN OF UNITY!
VARIABLE BOUNDS FACTOR = 10000.0

(no displacement limitations; no frequency limitations; no design variable linking)

If there is no design variable linking, the default values are

UNIFORM DV LOWER BOUNDS = 0.01
UNIFORM DV UPPER BOUNDS = 1.0E10

If there is design variable linking, the default values are

LINKING FACTORS OF UNITY!
UNIFORM LINKED DV LOW BOUNDS = 0.01
UNIFORM LINKED DV UP BOUNDS = 1.0E10

CALCULATE INITIAL DESIGN! **ANLID!**

Description: This feature has not yet been implemented.

CALCULATE LINKING FACTORS! **ANLLFACT!**

Description: This optional keyword is used to request that the fixed factors (or ratios) which relate the areas (or thicknesses) of each element to their corresponding linked design variable be calculated automatically from the initial design.

Data value: No data follow this keyword (just enter the keyword alone).

Default: The linking factors are set to 1.0.

Example: **CALCULATE LINKING FACTORS!**

Remarks: If present, this keyword must be specified after DESIGN VARIABLE LINKING:. Also, this keyword requires INITIAL DESIGN FOLLOWS: (which must precede this keyword) because the keyword CALCULATE INITIAL DESIGN! has not been implemented yet. It is not allowed with

INITIAL DESIGN OF UNITY!
LINKED INITIAL DESIGN:
LINKING FACTORS OF UNITY!
LINKING FACTORS FOLLOW:

The linking factors are calculated from the given initial design as follows: The element with the largest initial area (or thickness) within each linking group is chosen as the "representative" element for that design variable group. The factor for each element within the group is then fixed as the ratio of its initial area (or thickness) to the initial area (or thickness) of the representative element.

DISPLACEMENTS IN FEET! **DISPUFT!**

Description: This optional keyword is used to specify that the displacement limitation is given in feet.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Displacement units are in inches.

Example: **DISPLACEMENTS IN FEET!**

DISPLACEMENTS IN INCHES! **DISPUIN!**

Description: This optional keyword is used to specify that the displacement limitation is given in inches.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Displacement units are in inches.

Example: **DISPLACEMENTS IN INCHES!**

DISCRETE DV LOWER BOUNDS: **DVLBD:**

Description: This optional keyword specifies the lower bound for the design variable associated with each element.

Subsequent data line: Contains real numbers, the lower bound for each and every design variable (1 to *N*).

Reals 1 to N: Value for lower bound for each design variable

Default: The lower bounds are 0.01.

Example: **DISCRETE DV LOWER BOUNDS:**

0.001 0.001 0.001

(Here the lower bound for each of the three design variables is 0.001.)

Remarks: All bounds specified must be positive. This keyword should not be used when any one of these three keywords is present:

UNIFORM DV LOWER BOUNDS=
LINKED DV LOWER BOUNDS:
UNIFORM LINKED DV LOW BOUNDS=

The required unit of measure for this keyword is square inches or inches.

UNIFORM DV LOWER BOUNDS= **DVLBU=**

Description: This optional keyword is used to specify Uniform lower bounds for design variables.

Data value: Following the keyword, enter one real number.

Real 1: Value for uniform lower bound

Default: The lower bounds are 0.01.

Example: **UNIFORM DV LOWER BOUNDS = 0.001**

(Here the uniform lower bound for all design variables is 0.001.)

Remarks: The bound specified must be positive. This keyword should not be used when any one of these three keywords is present:

DISCRETE DV LOWER BOUNDS:
LINKED DV LOWER BOUNDS:
UNIFORM LINKED DV LOW BOUNDS=

The required unit of measure for this keyword is square inches or inches.

DISCRETE DV UPPER BOUNDS: **DVUBD:**

Description: This optional keyword is used to specify the upper bound for the design variable.

Subsequent data line: Contains real numbers, the upper bound for each and every design variable (1 to *N*).

Reals 1 to N: Value for upper bound for each design variable

Default: The upper bounds are 1.0×10^{10}.

Example: **DISCRETE DV UPPER BOUNDS:**

1.D4 1.D4 1.D4

(Here the upper bound for each of the three design variables is 10 000.0.)

Remarks: All bounds specified must be positive. This keyword should not be used when any one of these three keywords is present:

UNIFORM DV UPPER BOUNDS=
LINKED DV UPPER BOUNDS:
UNIFORM LINKED DV UP BOUNDS=

The required unit of measure for this keyword is square inches or inches.

UNIFORM DV UPPER BOUNDS= **DVUBU=**

Description: This optional keyword is used to specify uniform upper bounds for design variables.

Data value: Following the keyword, enter one real number.

Real 1: Value for uniform upper bound

Default: The upper bounds are 1×10^{10}.

Example: **UNIFORM DV UPPER BOUNDS = 1.0D4**

(Here the uniform upper bound for all design variables is 10 000.0.)

Remarks: The bound specified must be positive. This keyword should not be used when any one of these three keywords is present:

DISCRETE DV UPPER BOUNDS:
LINKED DV UPPER BOUNDS:
UNIFORM LINKED DV UP BOUNDS=

The required unit of measure for this keyword is square inches or inches.

FREQUENCY LIMITS: **FREQL:**

Description: This optional keyword is used to specify the natural frequency limitations for each mode.

Subsequent data lines: Contain an integer number and a real number (one set per line). Enter zeros to end this input.

Integer 1: Frequency mode number
Real 1: Value of frequency limitation in hertz

To terminate this keyword, enter two zeros (one integer and one real) on the last line.

Default: No frequency constraints are included.

Note: The maximum number of frequency constraints allowed is three.

Example: **FREQUENCY LIMITS:**

1 14.142136
0 0.0

(Here a single frequency constraint is specified; the fundamental (mode 1) frequency limitation is 14.142136.)

Remarks: Frequency constraints are not allowed unless the keyword BEGIN DYNAMIC ANALYSIS PARMS! has been given in the ANLDAT file. If the mode numbers that are to be calculated were specified in the analysis data file, ANLDAT, the mode numbers specified here must also have been specified there.

The required unit of measure for this keyword is hertz.

INITIAL DESIGN FOLLOWS: **GIVID:**

Description: This optional keyword is used to specify the initial values for each element area (or thickness).

Subsequent data line: Contains real numbers, the initial value for the area (or thickness) of each element (1 to *N*).

Reals 1 to N: Initial value for each element area (or thickness)

Default: Initial values for all design variables are 1.0.

Example: **INITIAL DESIGN FOLLOWS:**

2.000 2.000 2.000

(Here there are three elements whose initial areas are set to 2.000.)

Remarks: All initial values specified must be positive. This keyword is allowed with CALCULATE LINKING FACTORS!. It is not allowed with

LINKING FACTORS OF UNITY!
LINKING FACTORS FOLLOW:
INITIAL DESIGN OF UNITY!
LINKED INITIAL DESIGN:
CALCULATE INITIAL DESIGN!

If CALCULATE LINKING FACTORS! is present, INITIAL DESIGN FOLLOWS: should be given before CALCULATE LINKING FACTORS! keyword.

The required unit of measure for this keyword is square inches or inches.

LINKING FACTORS FOLLOW: **GIVLFACT:**

Description: This optional keyword is used to specify the fixed factors (or ratios) that relate the areas (or thicknesses) of each element to their corresponding linked design variable (see the keyword LINKDV:).

Subsequent data lines: Contain integer and real numbers as shown below. Repeat on subsequent lines as needed. Enter a zero on the last line to end this keyword.

Integer 1: Total number of elements whose factors are to be specified on this line *N*
Integer 2: Element number
Real 1: Linking factor

Integer 2 and real 1 are repeated *N* times. To terminate this input, enter a zero (integer) on the last line.

Default: A linking factor of 1.0 is assigned to members not listed.

Example: **LINKING FACTORS FOLLOW:**

1 3 0.50
0

(Here only one factor is specified. The area of element 3 will be 0.5 times the value of its corresponding linked design variable. The linking factors for all other elements will be 1.0.)

Remarks: All linking factors specified must be in the interval 0.0 to 1.0.This keyword is allowed with

INITIAL DESIGN OF UNITY!
LINKED INITIAL DESIGN:

It is not allowed with

INITIAL DESIGN FOLLOWS:
CALCULATE INITIAL DESIGN!
LINKING FACTORS OF UNITY!
CALCULATE LINKING FACTORS!

LINKED INITIAL DESIGN: **GIVLID:**

Description: This optional keyword is used to specify the initial values for the linked design variables.

Subsequent data line: Contains real numbers, the initial values for each linked design variable (1 to *N*).

Reals 1 to N: Initial values for each linked design variable

Default: Initial values for all design variables are 1.0.

Example: **LINKED INITIAL DESIGN:**

1.000 1.000 1.000

(Here there are three linked design variables, and their initial values are set to 1.000.)

Remarks: All initial values specified must be positive. If this keyword is present, it must be specified after DESIGN VARIABLE LINKING:. This keyword is allowed with

LINKING FACTORS OF UNITY!
LINKING FACTORS FOLLOW:

It is not allowed with

CALCULATE LINKING FACTORS!
INITIAL DESIGN OF UNITY!
INITIAL DESIGN FOLLOWS:
CALCULATE INITIAL DESIGN!

The required unit of measure for this keyword is square inches or inches.

LINKED DV LOWER BOUNDS: **LDVLB:**

Description: This optional keyword is used to specify the lower bounds for the linked design variables.

Subsequent data line: Contains real numbers, the lower bounds for the linked design variables (1 to *N*).

Reals 1 to N: Value of lower bounds for linked design variables

Default: The lower bounds are 0.01.

Example: **LINKED DV LOWER BOUNDS:**

0.001 0.001 0.001

(Here the lower bounds for the three linked design variables are 0.001.)

Remarks: All bounds specified must be positive. If this keyword is present, it must be specified after DESIGN VARIABLE LINKING:. This keyword should not be used when any one of these three keywords is present:

DISCRETE DV LOWER BOUNDS:
UNIFORM DV LOWER BOUNDS=
UNIFORM LINKED DV LOW BOUNDS=

The required unit of measure for this keyword is square inches or inches.

UNIFORM LINKED DV LOW BOUNDS= **LDVLBU=**

Description: This optional keyword is used to specify the uniform lower bounds for the linked design variables.

Data value: Following the keyword, enter a real number.

Real 1: Value of uniform lower bounds for linked design variables

Default: The lower bounds are 0.01.

Example: **UNIFORM LINKED DV LOW BOUNDS = 0.02**

(Here the uniform lower bounds for the linked design variables are 0.02.)

Remarks: The bounds specified must be positive. This keyword should not be used when any one of these three keywords is present:

DISCRETE DV LOWER BOUNDS:
UNIFORM DV LOWER BOUNDS=
LINKED DV LOWER BOUNDS:

The required unit of measure for this keyword is square inches or inches.

LINKED DV UPPER BOUNDS: **LDVUB:**

Description: This optional keyword is used to specify the upper bounds for the linked design variables.

Subsequent data line: Contains real numbers, the upper bounds for the linked design variables (1 to *N*).

Reals 1 to N: Value of upper bounds for linked design variables

Default: The upper bounds are 1×10^{10}.

Example: **LINKED DV UPPER BOUNDS:**

1.D5 1.D5 1.D5

(Here the upper bounds for the three linked design variables are 1×10^{5}.)

Remarks: All bounds specified must be positive. If this keyword is present, it must be specified after DESIGN VARIABLE LINKING:. This keyword should not be used when any one of these three keywords is present:

DISCRETE DV UPPER BOUNDS:
UNIFORM DV UPPER BOUNDS=
UNIFORM LINKED DV UP BOUNDS=

The required unit of measure for this keyword is square inches or inches.

UNIFORM LINKED DV UP BOUNDS= **LDVUBU=**

Description: This optional keyword is used to specify the uniform upper bounds for the linked design variables.

Data value: Following the keyword, enter a real number.

Real 1: Value of uniform upper bounds for linked design variables

Default: The upper bounds are 1×10^{10}.

Example: **UNIFORM LINKED DV UP BOUNDS = 1.0D4**

(Here the uniform upper bounds for the linked design variables are 10 000.0.)

Remarks: The bounds specified must be positive. This keyword should not be used when any one of these three keywords is present:

DISCRETE DV UPPER BOUNDS:
UNIFORM DV UPPER BOUNDS=
LINKED DV UPPER BOUNDS:

The required unit of measure for this keyword is square inches or inches.

DESIGN VARIABLE LINKING: **LINKDV:**

Description: This optional keyword is used to indicate that linking of design variables will be specified.

Subsequent data lines: Contain integer numbers, as given below, for each design variable group, one per line. Enter a zero to end this input.

Integer 1: Number of elements to be linked in design variable group *N*
Integers 2 to N+1: Element numbers

To terminate this input, enter one zero (integer) on the last line.

Default: Each element is assigned its own design variable (i.e., there is no design variable linking).

Example: **DESIGN VARIABLE LINKING:**

2 1 5
2 2 4
1 3
0

(Here three linked design variables result from grouping elements 1 and 5 together, grouping elements 2 and 4 together, and allowing element 3 to form a group on its own. See figure 7 on page 144.)

Remarks: Each element must be given in one and only one set.

DISP LIMITS BY DIRECTION: **STIFFD:**

Description: This optional keyword is used to indicate that the displacement limits are to be imposed at all nodes, in each of the two or three directions.

Subsequent data line: Contains *N* real numbers, where *N* is the dimension of the problem (2 or 3).

Reals 1 to N: Displacement limits

Default: No displacement constraints are included.

Example: **DISP LIMITS BY DIRECTION:**

0.2 0.05

(Here for this two-dimensional problem, displacements are limited at all nodes by 0.2 in. in the *X* direction and 0.05 in. in the *Y* direction).

Remarks: All displacement limitations specified must be positive. The default unit of measure for this keyword is inches.

DISP LIMITS BY NODES: **STIFFN:**

Description: This optional keyword is used to specify the nodal displacement limits.

Subsequent data lines: Contain two integers and one real number (per line) as given below. The direction is indicated by the integers 1, 2, and 3 corresponding to the *X*, *Y*, and *Z* directions, respectively. Enter zero values to end this input.

Integer 1: Node number
Integer 2: Direction of displacement
Real 3: Value of displacement (>0)

To terminate this keyword enter two integer zeros and one real zero on the last line.

Default: No displacement constraints are included.

Note: The maximum number of displacement constraints allowed is 50.

Example: **DISP LIMITS BY NODES:**

1 1 0.2
1 2 0.05
0 0 0.0

(Here two displacement constraints are specified at node 1: 0.2-in. limit in the *X* direction and 0.05 in. in the *Y* direction.)

Remarks: All displacement limitations specified must be positive. The default unit of measure for this keyword is inches.

STRESS LIMITS: **STRENGTH:**

Description: This optional keyword is used to specify the allowable stresses: tension, compression, and shear (for each material).

Subsequent data lines: Contain three real numbers (one line per material).

Real 1: Stress allowable in tension
Real 2: Stress allowable in compression
Real 3: Stress allowable in shear

Default: 1.0E+10 1.0E+10 1.0E+10

Example: **STRESS IN PSI!**
STRESS LIMITS:

0.1000E+14 0.1000E+14 0.1000E14

(Here the stress limitations for the single material are given as artificially high numbers so that no stress constraints will become active.)

Remarks: All stress allowables specified must be positive. The default unit of measure for this keyword is kips per square inch.

STRESS IN KSI! **STRUKSI!**

Description: This optional keyword is used to specify that stress units are given in kips per square inch.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Stress units are in kips per square inch.

Example: **STRESS IN KSI!**

STRESS IN PSI! **STRUPSI!**

Description: This optional keyword is used to specify that stress units are given in pounds per square inch.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Stress units are in kips per square inch.

Example: **STRESS IN PSI!**

INITIAL DESIGN OF UNITY! **UNITID!**

Description: This optional keyword is used to indicate that initial values of unity are used for the design variables.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Initial values for all design variables are 1.0.

Example: **INITIAL DESIGN OF UNITY!**

Remarks: This keyword is allowed with LINKING FACTORS OF UNITY! or with LINKING FACTORS FOLLOW:. It is not allowed with

CALCULATE LINKING FACTORS!
LINKED INITIAL DESIGN:
INITIAL DESIGN FOLLOWS:
CALCULATE INITIAL DESIGN!

LINKING FACTORS OF UNITY! **UNITLFACT!**

Description: This optional keyword is used to specify that the fixed factor (or ratio) which relates the area (or the thickness) of each element to its corresponding linked design variable is unity.

Data value: No data follow this keyword (just enter the keyword alone).

Default: The linking factors are 1.0.

Example: **LINKING FACTORS OF UNITY!**

Remarks: This keyword is allowed with

INITIAL DESIGN OF UNITY!
LINKED INITIAL DESIGN:

It is not allowed with

INITIAL DESIGN FOLLOWS:
CALCULATE INITIAL DESIGN!
LINKING FACTORS FOLLOW:
CALCULATE LINKING FACTORS!

VARIABLE BOUNDS FACTOR= **VBMAX=**

Description: This optional keyword is used to specify the factor for the design variable bounds associated with move limits.

Data value: Following the keyword, enter a real number.

Real 1: Value for variable bounds factor

Default: The factor for the design variable bounds is 10 000.0.

Example: **VARIABLE BOUNDS FACTOR = 100.0**

(Here the factor for the design variable bounds is set to 100.0.)

Remarks: The bounds factor specified must be positive. Design variable lower bounds are taken as the larger of the lower bounds specified elsewhere by the user and the initial design variables divided by the variable bounds factor. Similarly, upper bounds are taken as the smaller of the upper bounds specified and the product of the initial design variables with the variable bounds factor. In other words, move limits are enforced on the initial design.

3.3 Optimization Data Files

The third input file required by CometBoards is an optimization file. This file is specific to a particular optimizer and contains optimization parameters, such as convergence criteria, maximum iterations, penalty function parameters, and update methods. There is one optimization data file for each of seven different optimizers:

(1) SUMTDAT, used with SUMT
(2) SLPDAT, used with SLP
(3) FDDAT, used with FD
(4) IMSLDAT, used with IMSL
(5) SQPDAT, used with SQP
(6) OCDAT, used with OC
(7) FUDDAT, used with FUD

Like the previously described input files, these files also contain short names or long names for the keywords, and they may be entered in mixed, upper, or lower case.

In the next sections the data file for SUMT is described, followed by the SLP and FD data files, the IMSL data file, the SQP file, and finally the OC and FUD data files.

3.3.1 Optimization Data File SUMTDAT

This section describes the optimization data file required when the NEWSUMT optimization technique is used. It consists of the 11 keywords depicted in table 5. The descriptions of the keywords on subsequent pages are arranged alphabetically by their short names. Each keyword (short name or long name) may be entered in mixed case (upper or lower), as given. All keywords are optional and the order in which they are given is irrelevant with the exception of PENALTY PARMS FOLLOW:. It should be specified before CALCULATE PENALTY PARM! (if specified at all).

Default values have been given to all keywords in SUMTDAT and are listed in table 5.

TABLE 5.—TYPICAL KEYWORD SEQUENCE IN SUMTDAT

Short name	Long name	Default
IPRINT=	I PRINT FLAG=	0 (initial and final designs)
IGCTRL=	I GRADIENT CONTROL=	All gradients supplied by analyzer
IGCTRL:	I GRADIENT CONTROL:	All gradients supplied by analyzer
ILIM:	ITERATION LIMITS:	20 6 15
STOPC:	STOP CRITERIA:	0.001 0.001 0.005
STEPMX=	STEP LENGTH MAX=	2.0
GIVPEN:	PENALTY PARMS FOLLOW:	Initial penalty parameter to be calculated; others are 0.1 and 1.E–13.
ANLPEN!	CALCULATE PENALTY PARM!	Yes
NOBJ!	NONLINEAR OBJ!	Linear objective
LCON:	LINEAR CON:	Nonlinear constraints
IPSC:	I PARTIAL SIDE CON:	3 (for both bounds)

CALCULATE PENALTY PARM! **ANLPEN!**

Description: This optional keyword requests that NEWSUMT calculate the initial value for the penalty multiplier.

Data value: No data follow this keyword (just enter the keyword alone).

Default: NEWSUMT calculates the initial penalty multiplier unless the keyword PENALTY PARMS FOLLOW: is present. In this latter case the initial penalty multiplier supplied with that keyword is used.

Example: **CALCULATE PENALTY PARM!**

Remarks: This keyword should follow PENALTY PARMS FOLLOW: in order to be effective.

PENALTY PARMS FOLLOW: **GIVPEN:**

Description: This optional keyword is used to change the default values for the following penalty multipliers:

Penalty multiplier	Default
Initial	1.0
Decrease ratio	0.1
Lower bound for	1×10^{-13}

Subsequent data line: Contains the three new real values for these parameters.

Real 1: Initial value for penalty multiplier
Real 2: Penalty multiplier decrease ratio
Real 3: Lower bound for penalty multiplier

Default: 1.0 0.1 1×10^{-13} (see above)

Example: **PENALTY PARMS FOLLOW:**

0.001 0.01 1.E–8

I GRADIENT CONTROL= **IGCTRL=**

Description: This keyword is used to request that the gradients of the objective function and/or constraints are to be approximated by NEWSUMT.

Data value: Following the keyword, enter one positive integer.

Integer 1:
1 For approximation of objective function
2 For approximation of all constraints
3 For approximation of nonlinear constraints only
4 1 and 2 combined
5 1 and 3 combined

Default: All gradients are supplied by the analyzer.

Example: **I GRADIENT CONTROL = 4**

(Here NEWSUMT will generate all gradients, using the default of 0.01 as the finite difference step length.)

Remarks: When an equal sign follows the keyword, use the default finite difference step length of 0.01. In order to specify the finite difference step length, use this keyword followed by a colon (see IGCTRL: keyword).

I GRADIENT CONTROL: **IGCTRL:**

Description: Specify this keyword to request that the gradients of the objective function and/or constraints are to be approximated by NEWSUMT and to supply the finite difference step lengths.

Subsequent data line: Contains one positive integer (as described below) followed by real numbers that specify the finite difference step lengths for the design variables (1 to *N*).

Integer 1:
1 For approximation of objective function
2 For approximation of all constraints
3 For approximation of nonlinear constraints only
4 1 and 2 combined
5 1 and 3 combined

Reals 1 to N: Value for finite difference step length for each design variable

Default: All gradients are supplied by the analyzer.

Example: **I GRADIENT CONTROL:**

4 0.001 0.001 0.001

(Here NEWSUMT will generate all gradients, using 0.001 as the finite difference length size for the three design variables.)

Remarks: All step lengths specified must be positive. When a colon follows the keyword, the user must supply a finite difference step length for each design variable. To use the default finite difference step lengths, use this keyword followed by an equal sign (see IGCTRL= keyword).

Optimization Data File SUMTDAT

ITERATION LIMITS: **ILIM:**

Description: This optional keyword is used to change the default values of the maximum iterations allowed for the following:

Maximum iteration limit	Default
Golden section	20
One-dimensional searches	6
Unconstrained minimizations	15

Subsequent data line: Contains these three new integer numbers

Integer 1: Maximum number of golden section iterations
Integer 2: Maximum number of one-dimensional searches
Integer 3: Maximum number of unconstrained minimizations

Default: 20 6 15 (see above)

Example: **ITERATION LIMITS:**

30 30 30

Remarks: All iteration limits specified must be positive.

I PRINT FLAG= **IPRINT=**

Description: This optional keyword is used to specify the level of printed output from the optimizer.

Data value: Following the keyword, enter a nonnegative integer number (generally between zero and 4). The larger the number, the more output is printed out.

Integer 1: Number greater than or equal to zero

Default: Zero (Only the initial and final designs are printed by the optimizer.)

Example: **I PRINT FLAG = 2**

Remarks: The number specified must be nonnegative.

Optimization Data File SUMTDAT

I PARTIAL SIDE CON: **IPSC:**

Description: This optional keyword is used to remove restrictions of the upper or lower bounds on specific design variables.

Subsequent data lines: Contain a pair of integers for each design variable whose upper or lower bound is to be restricted.

Integer 1: Design variable number
Integer 2: 0 For no side constraints
1 Restriction on lower bounds only
2 Restriction on upper bounds only
3 Restriction on both bounds

The partial side constraints keyword is terminated with a pair of zeros on the last line.

Default: 3 (Both lower and upper bounds are restricted for all design variables not specified with this keyword.)

Example: **I PARTIAL SIDE CON:**

1 1
2 2
0 0

(Here two design variables are only partially restricted on bounds; design variable 1 is restricted at lower bounds only, and design variable 2 is restricted at upper bounds only.)

LINEAR CON: **LCON:**

Description: This optional keyword is used to indicate that the constraints are assumed to be linear.

Subsequent data line: Contains an integer number *N*, where *N* is the total number of linear constraints, followed by the constraint numbers (integers), which are assumed to be linear (1 to *N*).

Integer 1:	Total number of linear constraints (*N*)
Integers 2 to N+1:	Constraint numbers, which are assumed to be linear

Default: All constraints are assumed to be nonlinear.

Example: **LINEAR CON:**

3 1 2 3

(Here the first three constraints are assumed to be linear.)

Remarks: Use of this keyword is not recommended.

NONLINEAR OBJ! **NOBJ!**

Description: This optional keyword is used to indicate that the objective function is assumed to be nonlinear.

Data value: No data follow this keyword (just enter the keyword alone).

Default: The objective (merit) function is assumed to be linear.

Example: **NONLINEAR OBJ!**

STEP LENGTH MAX= **STEPMX=**

Description: This optional keyword is used to specify the maximum length of the initial step for one-dimensional searches.

Data value: Following the keyword, enter one real number for the maximum length.

Real 1: Maximum length of initial step for one-dimensional searches

Default: The maximum step length is 2.0.

Example: **STEP LENGTH MAX = 1.0**

STOP CRITERIA: **STOPC:**

Description: This optional keyword is used to change the default values for the following convergence criteria:

Convergence criterion	Default
Golden section	0.001
One-dimensional search	0.001
Overall	0.005

Subsequent data line: Contains the three new real numbers, one for each criterion listed above.

Real 1: Golden section convergence criterion
Real 2: One-dimensional search convergence criterion
Real 3: Overall convergence criterion

Default: 0.001 0.001 0.005 (see above)

Example: **STOP CRITERIA:**

0.001 0.001 0.001

3.3.2 Optimization Data Files SLPDAT and FDDAT

This section describes the optimization data file required when the SLP or FD optimization technique is used. The file consists of the four keywords depicted in table 6. The descriptions of the keywords on subsequent pages are arranged alphabetically by their short names. Each keyword (short name or long name) may be entered in mixed, upper, or lower case. All keywords are optional and the order in which they are given is irrelevant. Default values have been assigned for all keywords. The keywords associated with these two files are exactly the same, with the exception noted below.

The choice of the following keywords depends on the optimization technique chosen. Use ITERATION LIMITS= with FD, and ITERATION LIMITS: with SLP.

Default values for keywords in SLPDAT and FDDAT are given in table 6.

TABLE 6.—TYPICAL KEYWORD SEQUENCE IN SLPDAT AND FDDAT

Short name	Long name	Default
JPRINT=	J PRINT FLAG=	1 (initial and final designs)
GCTRL!	GRADIENT CONTROL!	Gradients supplied by analyzer
ILIM:	ITERATION LIMITS:	20, 40 (for SLP)
ILIM=	ITERATION LIMITS=	40 (for FD)

GRADIENT CONTROL! **GCTRL!**

Description: This optional keyword is used to request that all gradients be generated by the optimizer.

Data value: No data follow this keyword (just enter the keyword alone).

Default: Gradients are supplied by the analyzer.

Example: **GRADIENT CONTROL!**

ITERATION LIMITS: **ILIM:**

Description: This optional keyword is used to change the following defaults for SLP:

Iteration limit	Default
Maximum number of linearized subproblems solved by SLP	20
Maximum number of iterations allowed to solve linearized subproblem for SLP	40

Subsequent data line: Contains the two new integer numbers specified above.

Integer 1: Maximum number of linearized subproblems
Integer 2: Maximum number of iterations

Default: 20 40 (see above)

Example: **ITERATION LIMITS:**

100 100

Remarks: Both limits specified must be positive. The use of this keyword is valid only with the SLP optimization technique. If FD is chosen as the optimizer, use the keyword ITERATION LIMITS= instead.

ITERATION LIMITS= **ILIM=**

Description: This optional keyword is used to change the following default for FD:

Iteration limit	Default
Maximum number of iterations for FD	40

Data value: Following the keyword, enter one integer number for the maximum number of iterations.

Integer 1: Maximum number of iterations

Default: 40 (see above)

Example: **ITERATION LIMITS = 100**

Remarks: The limit specified must be positive. The use of this keyword is valid only with the FD optimization technique. If the optimizer SLP is chosen, use the keyword ITERATION LIMITS: instead.

J PRINT FLAG= **JPRINT=**

Description: This optional keyword is used to specify the level of printed output from the optimizer.

Data value: Following the keyword, enter an integer number between zero and 3. The larger the number, the more output is printed.

Integer 1: Number between zero and 3

Default: 1 (Only the initial and final designs are printed by the optimizer.)

Example: **J PRINT FLAG = 2**

3.3.3 Optimization Data File IMSLDAT

This section describes the optimization data file required when the IMSL optimization technique is used. The file consists of the three keywords depicted in table 7. The descriptions of the keywords on subsequent pages are arranged alphabetically by their short names. Each keyword (short name or long name) may be entered in mixed, upper, or lower case. All keywords are optional and the order in which they are given is irrelevant. Default values have been assigned to all keywords as shown in table 7.

TABLE 7.—TYPICAL KEYWORD SEQUENCE IN IMSLDAT

Short name	Long name	Default
IPRINT=	I PRINT FLAG=	1 (final design)
FINITE!	FINITE DIFF GRAD!	Analyzer to use analytical gradients
ILIM=	ITERATION LIMITS=	100

FINITE DIFF GRAD! **FINITE!**

Description: This optional keyword requests that the analyzer use finite difference gradients.

Data value: No data follow this keyword (just enter the keyword alone).

Default: The analyzer is to use analytical gradients.

Example: **FINITE DIFF GRAD!**

ITERATION LIMITS= **ILIM=**

Description: This optional keyword is used to specify the maximum number of iterations allowed.

Data value: Following the keyword, enter the integer number specifying the maximum number of iterations.

Integer 1: Maximum number of iterations

Default: The maximum number of iterations allowed is 100.

Example: **ITERATION LIMITS = 50**

Remarks: The limit specified must be positive.

I PRINT FLAG= **IPRINT=**

Description: This optional keyword is used to specify the level of printed output from the optimizer.

Data value: Following the keyword, enter an integer number between zero and 3. The larger the number, the more output is printed out.

Integer 1: Value between zero and 3

Default: 1 (Only the final design is printed by the optimizer.)

Example: **I PRINT FLAG = 2**

3.3.4 Optimization Data File SQPDAT

This section describes the optimization data file required when the SQP optimization technique is used. The file consists of the four keywords depicted in table 8. The descriptions of the keywords on subsequent pages are arranged alphabetically by their short names. Each keyword (short name or long name) may be entered in mixed, upper, or lower case. All keywords are optional and the order in which they are given is irrelevant. Default values have been assigned to all keywords as shown in table 8.

TABLE 8.—TYPICAL KEYWORD SEQUENCE IN SQPDAT

Short name	Long name	Default
IPRINT=	I PRINT FLAG=	–2 (final design)
GCTRL=	GRADIENT CONTROL=	Gradients supplied by analyzer
ILIM=	ITERATION LIMITS=	100
STOPC:	STOP CRITERIA:	0.001 0.001 0.005

GRADIENT CONTROL= **GCTRL=**

Description: This optional keyword is used to request that the constraint gradients be generated by the optimizer and to specify the finite difference step length.

Data value: Following the keyword, enter a real number specifying the delta used for finite difference gradients.

Real 1: Finite difference step length

Default: The gradients are supplied by the analyzer.

Example: **GRADIENT CONTROL = 0.01**

Optimization Data File SQPDAT

ITERATION LIMITS= **ILIM=**

Description: This optional keyword is used to specify the maximum number of iterations allowed.

Data value: Following the keyword, enter an integer number specifying the maximum number of iterations.

Integer 1: Maximum number of iterations

Default: The maximum number of iterations allowed is 100.

Example: **ITERATION LIMITS = 50**

Remarks: The limit specified must be positive.

Optimization Data File SQPDAT

I PRINT FLAG= **IPRINT=**

Description: This keyword is used to specify the level of printed output from the optimizer.

Data value: Following the keyword, enter an integer number between (–3 and 2). The larger the number, the more output is printed.

Integer 1: Value between –3 and 2

Default: –2. (Only the final design is printed by the optimizer.)

Example: **I PRINT FLAG = –1**

STOP CRITERIA: **STOPC:**

Description: This optional keyword is used to change the default values for the following convergence criteria:

Convergence criterion	Default
Acceptable violation of constraints	0.001
Acceptable tolerance for convergence	0.001
Acceptable relative change in cost function	0.005

Subsequent data line: Contains the three new real numbers, one for each criterion listed above.

Real 1: Acceptable violation of constraints
Real 2: Acceptable tolerance of convergence
Real 3: Acceptable relative change in cost function

Default: 0.001 0.001 0.005 (see above)

Example: **STOP CRITERIA:**

0.002 0.002 0.01

3.3.5 Optimization Data Files OCDAT and FUDDAT

This section describes the optimization data file required when the OC or FUD optimization technique is used. The file consists of the 19 keywords depicted in table 9. The descriptions of the keywords on subsequent pages are arranged alphabetically by their short names. Each keyword (short name or long name) may be entered in mixed, upper, or lower case. All keywords are optional and the order in which they are given is irrelevant. Default values have been assigned to all keywords as shown in table 9.

TABLE 9.—TYPICAL KEYWORD SEQUENCE IN OCDAT AND FUDDAT

Short name	Long name	Default
FINITE!	FINITE DIFF GRAD!	Analyzer to use analytical gradients
ILIM:	ITERATION LIMITS:	100 100 100
STOPC:	STOP CRITERIA:	0.001 0.001 0.001
ITRJUMP=	ITERATION JUMP=	5
STRAT:	STRATEGIES:	3 1 1 (freq, disp+freq, stress+disp+freq)
LMWEIGHT:	LM METHOD WEIGHTS:	0.0 0.0 0.0 0.0 0.0 1.0
XWEIGHT:	X METHOD WEIGHTS:	0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
CTHICK:	CONSTRAINT THICK:	0.9 for all constraints
PTHICK:	ACTIVE CON FRACT:	0.0 for minimums; 1.0 for maximums
NTHICK:	ACTIVE CON NUMBER:	0 0 0 1.0 ALL ALL ALL ALL
ACTFEAS!	MAKE 1 ACTIVE CON!	Final design not rescaled
ILAMDA:	INITIAL LAGRANGE:	100.0 for all multipliers
ACLRN:	ACCELERATE PARMS:	0.5 2.0 1.0 1.0 0.9
OCX4F:	X METHOD 4 PARMS:	0.1 0.1 0.0 0.15
AVDSGN!	AVERAGE DESIGN!	Design not averaged
INFEAS!	ALLOW INFEASIBLE!	Feasibility of design enforced
NEGLM!	ALLOW NEG LM!	Multipliers forced to remain positive
NEGDLS!	ALLOW NEG SCALES!	Scaling factors forced to remain positive
CEXP:	CON EXPONENTS:	1.0 1.5 2.0 (not currently used)

Choosing optimality criteria update formulas.—There are eight optimality criteria methods for updating the design variables, X METHOD WEIGHTS: (1 to 8), and six methods for updating the Lagrange multipliers, LM METHOD WEIGHTS: (0 to 5). Valid combinations of these methods are given in table 10.

TABLE 10.—COMBINATIONS FOR OC METHODS

Lagrange multiplier method	Design variable method
0	8
1 to 4	1 to 3
5	4 or 5 to 7

In particular, Lagrange multiplier update method 0 may only be used with design variable update method 8 and vice versa (see keywords LM WEIGHT: and X WEIGHT: for a brief description of these two update methods). However, any (or all) of the Lagrange multiplier methods 1 to 4 may be used with any (or all) of design methods 1 to 3. Lagrange multiplier update formula 5 may be used with either design variable update formula 4 or with any (or all) of design update formulas 5 to 7.

In cases where more than one update formula may be chosen at the same time (e.g., any (or all) of the Lagrange multiplier methods 1 to 4 may be chosen if the design updates are 1 to 3), the user must decide whether to use one of those methods (e.g., Lagrange multiplier method 2) exclusively or to update the quantities by forming a weighted sum of the available methods (e.g., by taking the arithmetic mean of the four updates by choosing weights of 0.25 for each of methods 1 to 4). In the former case the keyword and subsequent data line for the Lagrange multiplier update formula would be

```
LM METHOD WEIGHTS:
0.0   0.0   1.0   0.0   0.0   0.0
```

Note that the first real value above refers to method 0. In the latter case, described previously, the line would be

```
LM METHOD WEIGHTS:
0.0   0.25   0.25   0.25   0.25   0.0
```

Finally, note that design variable update formula 4 may not be combined with any other design variable update formula.

In defining the update weights the values of the weights of the keyword LM METHOD WEIGHTS: must always add to 1.0. Similarly, the values given for X METHOD WEIGHTS: must add to 1.0. More examples are given in section 5.

Choosing fully utilized design method.—The following five keywords are associated with the FUDDAT file and are exactly the same as the keywords for the OCDAT files:

```
FINITE DIFF GRAD!
ITERATION LIMITS:
STOP CRITERIA:
ITERATION JUMP=
MAKE 1 ACTIVE CON!
```

Note that the keywords X METHOD WEIGHTS: and LM METHOD WEIGHTS:, if given, are ignored. In particular, the following lines are added to the bottom of the FUDDAT file:

```
$
$ The following lines added for FUD optimizer:
$
LM METHOD WEIGHTS:
1.0   0.0   0.0   0.0   0.0   0.0
X METHOD WEIGHTS:
0.0   0.0   0.0   0.0   0.0   0.0   0.0   1.0
```

Also, the optimization strategies by level for FUD are set to 1, 0, 0 (see keyword STRAT:). These strategy values are required. If other values are given, the program automatically resets them and prints a warning message.

Note that the classical fully stressed design (FSD) can be obtained by using the FUD optimization technique and specifying no displacement or no frequency constraints in DSGNDAT.

ACCELERATE PARMS: **ACLRN:**

Description: This optional keyword is used to modify the default values of the acceleration parameters listed below:

Acceleration parameter	Default
Initial Lagrange multiplier update parameter	0.5
Initial design variable update parameter	2.0
Factor used to modify Lagrange multiplier update parameter	1.0
Factor used to modify design variable update parameter	1.0
Factor used to reduce constraint thickness	0.9

Subsequent data line: Contains these five new real values.

Real 1: Initial Lagrange multiplier update parameter
Real 2: Initial design variable update parameter
Real 3: Factor used to modify Lagrange multiplier update parameter
Real 4: Factor used to modify design variable update parameter
Real 5: Factor used to reduce constraint thickness

Default: 0.5 2.0 1.0 1.0 0.9 (see above)

Example: **ACCELERATE PARMS:**

0.5 2.0 1.05 1.02 0.97

Remarks: All values specified must be positive. The initial Lagrange multiplier update parameter is identified by p_0 in equation (10) (see section 2.1.5). The factor used to reduce the parameter is identified by γ in that same equation (10). Similarly, the initial design variable update parameter and its reduction factor are identified in equation (11) by q_0 and $\hat{\gamma}$, respectively. The constraint thicknesses (see the keyword CTHICK:) are reduced at each iteration by the fifth acceleration parameter shown above.

MAKE 1 ACTIVE CON! **ACTFEAS!**

Description: This optional keyword is used to allow a final rescaling of the design to ensure that at least one constraint becomes active.

Data value: No data follow this keyword (just enter the keyword alone).

Default: No final rescale is performed. (Thus the constraints in the final design may all be passive.)

Example: **MAKE 1 ACTIVE CON!**

AVERAGE DESIGN! **AVDSGN!**

Description: This optional keyword is used to request that each iteration's resulting design becomes the average of that returned by the design update procedure and that obtained subsequent to the enforcement of feasibility.

Data value: No data follow this keyword (just enter the keyword alone).

Default: By default, whenever the design feasibility enforcement is requested (see the keyword ALLOW INFEASIBLE!), the design is rescaled after the design update procedure in order to ensure feasibility. This rescaled design becomes the design at the end of the current iteration.

Example: **AVERAGE DESIGN!**

Remarks: If the keyword ALLOW INFEASIBLE! is present, this keyword is effectively ignored.

CON EXPONENTS: **CEXP:**

Description: This optional keyword is not currently used.

CONSTRAINT THICK: **CTHICK:**

Description: This optional keyword is used to modify the default values for the initial constraint thickness for the following constraint types:

Constraint type	Default
Stress	0.9
Displacement	0.9
Frequency	0.9

Subsequent data line: Contains the three new real values for the constraint thicknesses associated with each constraint type.

Real 1: Initial constraint thickness for all stress constraints
Real 2: Initial constraint thickness for all displacement constraints
Real 3: Initial constraint thickness for all frequency constraints

Default: 0.9 0.9 0.9 (see above)

Example: **CONSTRAINT THICK:**

0.1 0.1 0.1

Remarks: All three thicknesses must lie in the interval 0.0 to 1.0.The constraint thickness, which is modified at each iteration by the fifth parameter of the keyword ACCELERATE PARMS:, is used to help determine which constraints are active. A particular constraint is “active” if its value is greater than the negative of the corresponding constraint thickness by type.

FINITE DIFF GRAD! **FINITE!**

Description: This optional keyword requests that the analyzer use finite difference gradients.

Data value: No data follow this keyword (just enter the keyword alone).

Default: The analyzer is to use analytical gradients.

Example: **FINITE DIFF GRAD!**

INITIAL LAGRANGE: **ILAMDA:**

Description: This optional keyword is used to specify the initial values of the Lagrange multipliers.

Subsequent data line: Contains real numbers specifying the initial value of the Lagrange multipliers associated with each constraint (1 to *N*).

Reals 1 to N: Values of initial Lagrange multipliers

Default: The initial value of the Lagrange multipliers is 100.0.

Example: **INITIAL LAGRANGE:**

200.0 200.0 200.0

(Here the initial Lagrange multiplier for each of the three constraints is set to 200.0.)

Remarks: All initial values specified must be positive. The required unit of measure for this keyword is pounds-force.

ITERATION LIMITS: **ILIM:**

Description: This optional keyword is used to specify the iteration limits for each of the three optimization strategy levels (see keyword STRAT:).

Subsequent data line: Contains the three new integer values, the maximum iterations at each strategy level.

Integer 1: Maximum iteration at strategy level 1
Integer 2: Maximum iteration at strategy level 2
Integer 3: Maximum iteration at strategy level 3

Default: The iteration limit is 100 for all three strategy levels.

Example: **ITERATION LIMITS:**

200 200 100

(Here the maximum iterations are 200 for level 1, 200 for level 2, and 100 for level 3.)

Remarks: All three limits must be positive.

ALLOW INFEASIBLE! **INFEAS!**

Description: This optional keyword is used to temporarily allow the optimizer to use any infeasible design it may encounter during the iteration process.

Data value: No data follow this keyword (just enter the keyword alone).

Default: At the end of each iteration the design is rescaled to ensure that no constraints are violated.

Example: **ALLOW INFEASIBLE!**

Remarks: See also the keyword AVDSGN!.

ITERATION JUMP= **ITRJUMP=**

Description: This optional keyword is associated with comparisons made during checks for convergence. At the end of each iteration the values of the objective function, the design variables, and the Lagrange multipliers from the current iteration are compared with those of a previous iteration to determine whether the algorithm has converged. This keyword specifies which previous iteration to use. If the value specified is 1, the comparison is with the immediately prior iteration. If it is 2, the second prior iteration is used (i.e., the current iteration minus two), and so on.

Data value: Following the keyword, enter the new integer number.

Integer 1: Value for iteration jump

Default: The default value is 5.

Note: The maximum value that can be specified for this keyword is 10.

Example: **ITERATION JUMP = 10**

Remark: The value specified must be in the range 1 to 10.

LM METHOD WEIGHTS: **LMWEIGHT:**

Description: The Lagrange multipliers are updated by using one or more of the methods listed below. If more than one method is used, the updated multipliers are formed by a weighted average of the results of each method applied individually at the current iteration. This keyword is used to specify these weights. The methods and the defaults for their associated weights are

Method	Default
0—No Lagrange (classical FUD)	0.0
1—Linear form	0.0
2—Exponential form	0.0
3—Diagonalized Lagrange inverse form	0.0
4—Unrestricted Lagrange inverse form	0.0
5—Lagrange inverse form	1.0

Subsequent data line: Contains the six real values for these parameters.

Real 1: Weight associated with use of no Lagrange multiplier (classical FUD)
Real 2: Weight associated with linear form of Lagrange multiplier update
Real 3: Weight associated with exponential form of Lagrange multiplier update
Real 4: Weight associated with diagonalized Lagrange inverse form of Lagrange multiplier update
Real 5: Weight associated with unrestricted Lagrange inverse form of Lagrange multiplier update
Real 6: Weight associated with Lagrange inverse form of Lagrange multiplier update

Default: 0.0 0.0 0.0 0.0 0.0 1.0

Example: **LM METHOD WEIGHTS:**

0.0 1.0 0.0 0.0 0.0 0.0

(Here method 1 is used.)

Remarks: Note that the weights must each be in the interval 0.0 to 1.0 and collectively must sum to 1.0. Moreover, this keyword must be coupled with the keyword X METHOD WEIGHTS:. Refer to table 10 for the valid combinations of these methods. Note also that the linear form of Lagrange multiplier update (method 1) is only allowed with the truss (bar) element. See reference 5 for more details on the methods.

ALLOW NEG SCALES! **NEGDLS!**

Description: This optional keyword is used to allow the optimizer to use any negative scaling factors that may arise.

Data value: No data follow this keyword (just enter the keyword alone).

Default: By default the scaling factors for updates to the design are not allowed to become negative. At the end of the scaling routine any negative values are set to zero.

Example: **ALLOW NEG SCALES!**

ALLOW NEG LM! **NEGLM!**

Description: This optional keyword is used to allow the optimizer to use any negative Lagrange multipliers that may arise.

Data value: No data follow this keyword (just enter the keyword alone).

Default: By default the Lagrange multipliers are not allowed to become negative. At the end of the Lagrange multiplier update routine any negative values are set to zero.

Example: **ALLOW NEG LM!**

ACTIVE CON NUMBER: **NTHICK:**

Description: The set of "active" constraints is obtained by first determining the constraints that are within the constraint thickness (see the keyword CTHICK:) as well as those that are violated. The final number of active constraints is further modified by using this keyword and the keyword PTHICK:. This keyword is used to modify the default values for the minimum and maximum number of active constraints by type, as listed below:

Constraint type	Default
Minimum number of active stress constraints	0
Minimum number of active displacement constraints	0
Minimum number of active frequency constraints	0
Overall minimum number of all active constraints	1
Maximum number of active stress constraints	ALL
Maximum number of active displacement constraints	ALL
Maximum number of active frequency constraints	ALL
Overall maximum number of all active constraints	ALL

Subsequent data line: Contains the eight new integer values for these constraint types.

Integer 1: Minimum number of active stress constraints
Integer 2: Minimum number of active displacement constraints
Integer 3: Minimum number of active frequency constraints
Integer 4: Overall minimum number of all active constraints
Integer 5: Maximum number of active stress constraints
Integer 6: Maximum number of active displacement constraints
Integer 7: Maximum number of active frequency constraints
Integer 8: Overall maximum number of all active constraints

Default: 0 0 0 1 ALL ALL ALL ALL (see above)

Example: **ACTIVE CON NUMBER:**

1 1 1 3 3 1 1 5

Remarks: The minimum and maximum number of constraints specified by constraint type must be greater than or equal to zero and less than or equal to the total number of that type of constraint specified in DSGNDAT. The overall minimum and maximum must be greater than or equal to one and less than or equal to the total number of constraints altogether as specified in DSGNDAT.

For each constraint type (and overall for all constraint types taken together) the number of constraints chosen will not be less than the larger of the two minimums (as determined by the keywords ACTIVE CON FRACT: and ACTIVE CON NUMBER:; see the keyword PTHICK:) and not greater than the smaller of the two maximums.

X METHOD 4 PARMS: **OCX4F:**

Description: This optional keyword is used to change the default values for the fraction with which to weight the use of the scaling factors DLES (and the old design χ^{k-1}) during design variable updates when update method 4 (the melange form) is used, as given below:

Active constraint type	Element affected	Quantity used	Default fraction
Stress	Those with passive stress constraints	DLES	0.1
Displacement	All elements	DLES	0.1
Frequency	All elements	DLES	0.0
Displacement or frequency	Those with passive stress constraints	χ^{k-1}	0.15

Subsequent data line: Contains the four real values described above.

Real 1: Fraction with which to weight use of scaling factors for elements associated with passive stress constraints (used only when at least one stress constraint is active)

Real 2: Fraction with which to weight use of scaling factors when at least one displacement constraint is active

Real 3: Fraction with which to weight use of scaling factors when at lest one frequency constraint is active

Real 4: Fraction with which to weight values of design variables not currently associated with active stress constraint from the immediately prior iteration (used only when at least one displacement or frequency constraint is active)

Default: 0.1 0.1 0.0 0.15 (see above)

Example: **X METHOD 4 PARMS:**

0.1 0.1 0.1 0.1

(Here the weighting fraction for use of DLES and χ^{k-1} for each of these numbers is set to 0.1.)

Remarks: Each of the four numbers must be in the interval 0.0 to 1.0. See reference 5 for details.

ACTIVE CON FRACT: **PTHICK:**

Description: The set of “active” constraints is obtained by first determining the constraints that are within the constraint thickness (see the keyword CTHICK:) as well as those that are violated. The final number of active constraints is further modified by use of this keyword and the keyword NTHICK:. This keyword is used to modify the default values for the minimum and maximum fraction of the active constraints by type, as listed below:

Constraint type	Default
Minimum fraction of active stress constraints	0.0
Minimum fraction of active displacement constraints	0.0
Minimum fraction of active frequency constraints	0.0
Overall minimum fraction of all active constraints	0.0
Maximum fraction of active stress constraints	1.0
Maximum fraction of active displacement constraints	1.0
Maximum fraction of active frequency constraints	1.0
Overall maximum fraction of all active constraints	1.0

Subsequent data line: Contains the eight new real values for the constraint types listed above.

Real 1: Minimum fraction of active stress constraints
Real 2: Minimum fraction of active displacement constraints
Real 3: Minimum fraction of active frequency constraints
Real 4: Overall minimum fraction of all active constraints
Real 5: Maximum fraction of active stress constraints
Real 6: Maximum fraction of active displacement constraints
Real 7: Maximum fraction of active frequency constraints
Real 8: Overall maximum fraction of all active constraints

Default: 0.0 0.0 0.0 0.0 1.0 1.0 1.0 1.0 (see above)

Example: **ACTIVE CON FRACT:**

0.0 0.0 0.0 0.0 0.2 0.5 0.4 0.25

(If, for example, there were 20 stress constraints, 4 displacement constraints, and 1 frequency constraint within their constraint thicknesses—or which were violated—during a particular iteration, this keyword would require that a maximum of 4 stress, 2 displacement, and 1 frequency constraint be included in the active constraint set. Moreover, one of these constraints would have to be eliminated in order to produce a maximum of 6 active constraints overall. These maximums may be further modified by the keyword ACTIVE CON NUMBER:.)

Remarks: All eight values must lie in the interval 0.0 to 1.0. For each constraint type (and overall for all constraint types taken together), the number of constraints chosen will not be less than the larger of the two minimums (as determined by the keywords ACTIVE CON FRACT: and ACTIVE CON NUMBER:; see the keyword NTHICK:) and not greater than the smaller of the two maximums.

STOP CRITERIA: **STOPC:**

Description: This optional keyword is used to modify the default convergence criteria for the following parameters:

Parameter	Default
Design variables	0.001
Lagrange multipliers	0.001
Objective function	0.001

Subsequent data line: Contains the three new real values for the parameters specified above.

Real 1: Convergence criterion on relative change in L_2 norm of difference in design variables
Real 2: Convergence criterion on relative change in L_2 norm of difference in Lagrange multipliers
Real 3: Convergence criteria on relative change in objective function

Default: 0.001 0.001 0.001 (see above)

Example: **STOP CRITERIA:**

0.005 0.005 0.005

Remarks: All three values must be in the interval 0.0 to 1.0.

STRATEGIES: **STRAT:**

Description: Optimization proceeds in as many as three stages (or levels). At each level certain constraint types are addressed, depending on the strategy chosen, and the final design at this level is used as the initial design at the next level. This keyword is used to change the default optimization strategy levels.

The options for level one are

0 To skip this level
1 For consideration of stress constraints only
2 For consideration of displacement constraints only
3 For consideration of frequency constraints only

The options for level two are

0 To skip this level
1 For consideration of displacement and frequency constraints only
2 For consideration of stress and frequency constraints only
3 For consideration of stress and displacement constraints only

The options for level three are

0 To skip this level
1 For consideration of all constraints

Subsequent data line: Contains the three new integer values for the strategy option for each of these levels.

Integer 1: Strategy option for level one (0–3).
Integer 2: Strategy option for level two (0–3).
Integer 3: Strategy option for level three (0–1).

Default: The default options for each level are 3 1 1 (i.e., consideration of frequency constraints at level one, then consideration of stress and frequency constraints at level two, and finally consideration of all the constraints at level three).

Example: **STRATEGIES:**

1 2 0

(Here the optimization strategies are for stress constraints at level one and stress and frequency constraints at level two and skip level three.)

X METHOD WEIGHTS: **XWEIGHT:**

Description: The design variables are updated by using one or more of the methods listed below. If more than one method is used, the updated design is formed by a weighted average of the results of each method applied individually at the current iteration. This keyword is used to specify these weights. The methods and the defaults for their associated weights are as follows:

Method	Default
1–Exponential form	0.0
2–Linearized form	0.0
3–Reciprocal form	0.0
4–Melange form	1.0
5–Fully stressed design (FSD) plus exponential form	0.0
6–FSD plus linearized form	0.0
7–FSD plus reciprocal form	0.0
8–Fully utilized design (FUD)	0.0

Subsequent data line: Contains the eight real values for these parameters:

Real 1: Weight associated with exponential form of design variable update
Real 2: Weight associated with linearized form of design variable update
Real 3: Weight associated with reciprocal form of design variable update
Real 4: Weight associated with melange form of design variable update
Real 5: Weight associated with hybrid exponential form of design variable update
Real 6: Weight associated with hybrid linearized form of design variable update
Real 7: Weight associated with hybrid reciprocal form of design variable update
Real 8: Weight associated with fully utilized design form of design variable update

Default: 0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0 (see above)

Example: **X METHOD WEIGHTS:**

1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0

(Here method 1 is used.)

Remarks: Note that the weights must each be in the interval 0.0 to 1.0 and collectively must sum to 1.0. Moreover, this keyword must be coupled with the keyword LM METHOD WEIGHTS:. Refer to table 10 for the valid combinations of these methods. See reference 5 for more details on the methods.

3.4 Output User Information

The following information is produced on the standard output file:

(1) NASA DESIGN logo and optimization method used

(2) Echo of the anldat file (the analysis input data) as well as the formatted output of all parameters read in after data interpretation. (Note that the echo of the input in this file—and the two files below—will not be an exact character-by-character echo and in a few cases may produce more than one line of "echoed" data per line of input.)

(3) Echo of the dsgndat file (the design input data) as well as the formatted output of all parameters read in after data interpretation

(4) Echo of the optimization data file as well as the formatted output of optimization parameters read in

(5) Optimization control parameters and system parameters

(6) Iterative results depending on the print control parameter and optimization technique. These include the value of the objective function, the values of the design variables and constraints, dimensional searches, and Lagrange multipliers.

(7) When convergence, or the maximum number of iterations, is reached, the final results are printed out in this file, which depicts the following:

- (a) The optimization technique used
- (b) A limit message (if the maximum number of iterations is exceeded)
- (c) The title of the problem and the total number and type of constraints used
- (d) The value of the objective function at the optimal design
- (e) The total execution time in seconds
- (f) The number of iterations for convergence for each strategy level (FUD and OC only)
- (g) For each design variable the lower bounds, the initial design, the upper bounds, and the optimum design
- (h) For each constraint number the constraint value and whether it is an active or passive constraint
- (i) For FUD and OC methods the initial and final Lagrange multipliers for each constraint number
- (j) The total number of active stress constraints, displacement constraints, and/or frequency constraints
- (k) The total number of infeasible stress, displacement, and/or frequency constraints, if any.

When the execution fails, the program stops and an error message is provided by CometBoards.

It has been observed that when the IMSL optimization technique is used, quite often the final results described in (7) are not printed out. This is due to a fatal error from the IMSL subroutine DNCONG; however, the final convergence analysis is still provided by IMSL.

Section 4
Submitting a User-Defined Problem to CometBoards

In this section any words in **bold** should be typed exactly as shown. Any words in *italics* are parameters where a choice must be made by the user or a file name must be supplied. Finally, square brackets in command line syntax represent optional parameters and are not to be included in the command line.

Before submitting a problem to CometBoards the user must prepare three data files that define the analysis, design, and optimization parameters for the problem to be solved. To invoke CometBoards on any platform, type

optimize *optimizer analyzer* **>** *filename*

where **optimize** is the command that executes CometBoards

optimizer is

sumt	to use the sequence of unconstrained minimizations technique
slp	to use the sequential linear programming technique[1]
fd	to use the modified method of feasible directions[1]
imsl	to use the IMSL version of the successive quadratic programming technique[1]
sqp	to use the sequential quadratic programming technique of IDESIGN
oc	to use the optimality criteria methodology
fud	to use the fully utilized design capability

analyzer is

disp	to use the displacement analyzer
ifm	to use the integrated force method analyzer
sifms	to use the integrated force method analyzer with simplified stress gradients
sifmd	to use the integrated force method analyzer with simplified displacement gradients
sifmsd	to use the integrated force method analyzer with simplified stress and displacement gradients

filename is the file name (or on VM/CMS the file identification: filename filetype filemode) for the output file. The default is for output to go to the screen if the symbol > and the file name are missing.

After invocation the user will be prompted for the names of the three input data files.

For example to run an optimality criteria method with the displacement analyzer (and three data files named prob1.anldat, prob1.dsgndat, and prob1.ocdat) and to send the output to the file prob1.output, type the command

optimize oc disp > prob1.output

[1]The SLP, FD, and IMSL optimizers are not available at all sites.

Following each prompt enter the name of each data file:

prob1.anldat
prob1.dsgndat
prob1.ocdat

The prob1.anldat file includes the specification of the finite element model. The prob1.dsgndat file contains the design input data needed for constraint generation. The prob1.ocdat file specifies parameters for the optimizer, including which optimality criteria update formulas are to be used. For additional information on these files, see section 3.

4.1 Input File Specification Options

The names of the input data files may be specified directly on the command line by using the option flags **–anl**, **–dsgn**, and **–opt** as follows:

optimize *optimizer analyzer* **–anl** *anldat_file* **–dsgn** *dsgndat_file* **–opt** *optdat_file* > *filename*

where

anldat_file is the name of the anldat file, containing the finite element model specification
dsgndat_file is the name of the dsgndat file, containing the design input data
optdat_file is the name of the optdat file, containing parameters for the optimizer

and *optimizer*, *analyzer*, and *filename* are as described previously. Note that NO space is allowed after the hyphen in the option flags. Note also that the options may be given in any order, except that the optimizer and the analyzer, respectively, must be the first two arguments.

To use these options with the previous example, type the command

optimize oc disp –anl prob1.anldat –dsgn prob1.dsgndat –opt prob1.ocdat > prob1.output

If the file name options are specified, the user is not prompted for input data file names unless a specified file could not be found or could not be read.

On VM/CMS these options have slightly different syntax and defaults. In particular, each file is identified by a filename, a filetype, and a filemode. In each case the user may specify all three identifiers, the first two (in which case the filemode would become the default A), or the filename only (in which case both the filetype and the filemode are set to their default values, as indicated below). For the input data files, if no file identifier is specified or if an error is detected, the user will be prompted for the file identification. If no output file name is specified in the presence of the symbol >, execution will terminate prematurely, with an error message.

To run cometboards with these options on VM/CMS, type

optimize *optimizer analyzer* **–anl** *anldat_fn anldat_ft anldat_fm*
–dsgn *dsgndat_fn dsgndat_ft dsgndat_fm* **–opt** *optdat_fn*
optdat_ft optdat_fm > *fn ft fm*

where

anldat_fn is the filename of the anldat file, containing finite element model specification
anldat_ft is the filetype of the anldat file (default is ANLDAT)
anldat_fm is the filemode of the anldat file (default is A)
dsgndat_fn is the filename of the dsgndat file, containing design input data

dsgndat_ft is the filetype of the dsgndat file (default is DSGNDAT)
dsgndat_fm is the filemode of the dsgndat file (default is A)
optdat_fn is the filename of the optdat file, containing parameters for the optimizer
optdat_ft is the filetype of the optdat file (default is obtained by appending DAT to name of optimizer; e.g., SUMTDAT, SQPDAT, etc.)
optdat_fm is the filemode of the optdat file (default is A)
fn is the filename of the output file
ft is the filetype of the output file (default is OUT)
fm is the filemode of the output file (default is A)

4.2 Secondary Output File Retention Options

Certain types of output, which are normally discarded at the end of a run, may be retained with the use of the option flags –wt, –t, and/or –v. In each case the flag may be followed by the file name (or the file identification on VM/CMS) into which the output is to be placed. The first option (–wt) requests retention of unit 8 output, which records the number of times the analyzer is called, along with the structural weight for each design passed into the analyzer. The second option (–t) requests retention of unit 10 output, which records the cumulative CPU time, along with the structural weight at each call to the analyzer. The third option (–v) requests retention of unit 11 output, which records the values of the design variables, objective functions, and constraints and their gradients at each call to the analyzer. Note that this last file can grow quite large for large problems.

To run CometBoards and retain this additional output, type

optimize *optimizer analyzer* **–wt** *out8_file* **–t** *out10_file* **–v** *out11_file*

where

out8_file is the name of the file to contain weight-versus-reanalysis-call output (default, if option flag is included but file name is not present, is nasdes.wt_iter)
out10_file is the name of the file to contain weight-versus-CPU-time output (default, if option flag is included but file name is not present, is nasdes.wt_time)
out11_file is the name of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default, if option flag is included but file name is not present, is nasdes.verbose)

and *optimizer* and *analyzer* are as described previously. After invocation the user will be prompted for the names of the three input data files.

On VM/CMS type

optimize *optimizer analyzer* **–wt** *out8_fn out8_ft out8_fm* **–t** *out10_fn out10_ft out10_fm*
–v *out11_fn out11_ft out11_fm*

where

out8_fn is the filename of the file to contain weight-versus-reanalysis-call output (default is NASDES)
out8_ft is the filetype of the file to contain weight-versus-reanalysis-call output (default is WT_ITER)
out8_fm is the filemode of the file to contain weight-versus-reanalysis-call output (default is A)
out10_fn is the filename of the file to contain weight-versus-CPU-time output (default is NASDES)
out10_ft is the filetype of the file to contain weight-versus-CPU-time output (default is WT_TIME)
out10_fm is the filemode of the file to contain weight-versus-CPU-time output (default is A)

out11_fn is	the filename of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default is NASDES)
out11_ft is	the filetype of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default is VERBOSE)
out11_fm is	the filemode of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default is A)

After invocation the user will be prompted for the names of the three input data files.

4.3 Complete Command Syntax

In summary, the complete syntax of the shell script to run CometBoards on a Posix-based system is

optimize *optimizer analyzer* [**–anl** *anldat_file*] [**–dsgn** *dsgndat_file*] [**–opt** *optdat_file*]
[**–wt** [*out8_file*]] [**–t** [*out10_file*]] [**–v** [*out11_file*]] [> *filename*]

or, to run the demo

optimize *optimizer analyzer* **–demo** [*problem constraint(s)*]
[**–wt** [*out8_file*]] [**–t** [*out10_file*]] [**–v** [*out11_file*]] [> *filename*]

where *optimizer* is

sumt	to use the sequence of unconstrained minimizations technique
slp	to use the sequential linear programming technique[1]
fd	to use the modified method of feasible directions[1]
imsl	to use the IMSL version of the successive quadratic programming technique[1]
sqp	to use the sequential quadratic programming technique of IDESIGN
oc	to use the optimality criteria methodology
fud	to use the fully utilized design capability

analyzer is

disp	to use the displacement analyzer
ifm	to use the integrated force method analyzer
sifms	to use the integrated force method analyzer with simplified stress gradients
sifmd	to use the integrated force method analyzer with simplified displacement gradients
sifmsd	to use the integrated force method analyzer with simplified stress and displacement gradients

anldat_file is	the name of the anldat file, containing the finite element model specification
dsgndat_file is	the name of the dsgndat file, containing the design input data
optdat_file is	the name of the optdat file, containing parameters for the optimizer

out8_file is	the name of the file to contain weight-versus-reanalysis-call output (default, if option flag is included but file name is not present, is nasdes.wt_iter)
out10_file is	the name of the file to contain weight-versus-CPU-time output (default, if option flag is included but file name is not present, is nasdes.wt_time)

[1]The SLP, FD, and IMSL optimizers are not available at all sites.

out11_file is the name of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default, if option flag is included but file name is not present, is nasdes.verbose)

filename is the file name for the output file (default is for output to go to screen, if symbol > and file name are missing)

problem is

3link for a three-bar truss with linking and displacement constraints only
3 for a three-bar truss with stress, displacement, and/or frequency constraints
5sub1 for a five-bar truss (top loaded) with displacement constraints only
5sub2 for a five-bar truss (bottom loaded) with displacement constraints only
5sub3 for a five-bar truss (stretch loaded) with displacement constraints only
5sub4 for a five-bar truss (sway loaded) with displacement constraints only
5 for a five-bar truss with stress, displacement, and/or frequency constraints
10 for a 10-bar truss with stress, displacement, and/or frequency constraints
ring for a 60-bar trussed ring with 24 triangular elements and stress, displacement, and/or frequency constraints
fsw for a 135-member, forward-swept wing with stress and/or displacement constraints

constraint(s) consist of the following constraint types (alone or in any combination):

stress to include stress constraints
disp to include displacement constraints
freq to include frequency constraints

Note that if the –demo option is present, any of the three input file name options –anl, –dsgn, and –opt will be ignored. See section 6 for a complete description of the –demo option.

On VM/CMS the syntax is

optimize *optimizer analyzer* [**–anl** *anldat_fn* [*anldat_ft* [*anldat_fm*]]] [**–dsgn** *dsgndat_fn* [*dsgndat_ft* [*dsgndat_fm*]]][**–opt** *optdat_fn* [*optdat_ft* [*optdat_fm*]]] [**–wt** [*out8_fn* [*out8_ft* [*out8_fm*]]]] [**–t** [*out10_fn* [*out10_ft* [*out10_fm*]]]] [**–v** [*out11_fn* [*out11_ft* [*out11_fm*]]]] [> *fn* [*ft* [*fm*]]]

or, to run the demo

optimize *optimizer analyzer* **–demo** [*problem constraint(s)*] [**–wt** [*out8_fn* [*out8_ft* [*out8_fm*]]]] [**–t** [*out10_fn* [*out10_ft* [*out10_fm*]]]] [**–v** [*out11_fn* [*out11_ft* [*out11_fm*]]]] [> *fn* [*ft* [*fm*]]]

where *optimizer* is

sumt to use the sequence of unconstrained minimizations technique
slp to use the sequential linear programming technique[1]
fd to use the modified method of feasible directions[1]
imsl to use the IMSL version of the successive quadratic programming technique[1]

[1]The SLP, FD, and IMSL optimizers are not available at all sites.

sqp	to use the sequential quadratic programming technique of IDESIGN
oc	to use the optimality criteria methodology
fud	to use the fully utilized design capability

analyzer is

disp	to use displacement analyzer
ifm	to use integrated force method analyzer
sifms	to use integrated force method analyzer with simplified stress gradients
sifmd	to use integrated force method analyzer with simplified displacement gradients
sifmsd	to use integrated force method analyzer with simplified stress and displacement gradients

anldat_fn is	the filename of the anldat file, containing finite element model specification
anldat_ft is	the filetype of the anldat file (default is ANLDAT)
anldat_fm is	the filemode of the anldat file (default is A)
dsgndat_fn is	the filename of the dsgndat file, containing the design input data
dsgndat_ft is	the filetype of the dsgndat file (default is DSGNDAT)
dsgndat_fm is	the filemode of the dsgndat file (default is A)
optdat_fn is	the filename of the optdat file, containing parameters for the optimizer
optdat_ft is	the filetype of the optdat file (default is obtained by appending DAT to the name of the optimizer; e.g., SUMTDAT, SQPDAT, etc.)
optdat_fm is	the filemode of the optdat file (default is A)
out8_fn is	the filename of the file to contain weight-versus-reanalysis-call output (default is NASDES)
out8_ft is	the filetype of the file to contain the weight-versus-reanalysis-call output (default is WT_ITER)
out8_fm is	the filemode of the file to contain weight-versus-reanalysis-call output (default is A)
out10_fn is	the filename of the file to contain weight-versus-CPU-time output (default is NASDES)
out10_ft is	the filetype of the file to contain weight-versus-CPU-time output (default is WT_TIME)
out10_fm is	the filemode of the file to contain weight-versus-CPU-time output (default is A)
out11_fn is	the filename of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default is NASDES)
out11_ft is	the filetype of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default is VERBOSE)
out11_fm is	the filemode of the file to contain verbose information on design, objective, and constraints at each call to the analyzer (default is A)
fn is	the filename of the output file
ft is	the filetype of the output file (default is OUT)
fm is	the filemode of the output file (default is A)

problem is

3link	for a three-bar truss with linking and with displacement constraints only
3	for a three-bar truss with stress, displacement, and/or frequency constraints
5sub1	for a five-bar truss (top loaded) with displacement constraints only
5sub2	for a five-bar truss (bottom loaded) with displacement constraints only
5sub3	for a five-bar truss (stretch loaded) with displacement constraints only
5sub4	for a five-bar truss (sway loaded) with displacement constraints only
5	for a five-bar truss with stress, displacement, and/or frequency constraints
10	for a 10-bar truss with stress, displacement, and/or frequency constraints
ring	for a 60-bar trussed ring with 24 triangular elements and stress, displacement, and/or frequency constraints

	fsw	for a 135-member, forward-swept wing with stress and/or displacement constraints

constraint(s) consist of the following constraint types (alone or in any combination):

stress	to include stress constraints
disp	to include displacement constraints
freq	to include frequency constraints

Note that if the –demo option is present, any of the three input filename options –anl, –dsgn, and –opt will be ignored. See section 6 for a complete description of the –demo option.

Two brief forms of on-line help are available. If "optimize" is invoked with no parameters, a usage line specifying the order of the parameters is displayed. For a brief description of all parameters, type "optimize help."

Demonstration problems are available with CometBoards. These problems and their access are described in section 6.

The user may set up script files on Posix-based workstations or EXEC files on VM/CMS to run CometBoards in batch mode. These two files are given next for the example shown above. In each case two alternatives are provided.

(1) Batch file 1. C-Shell script for Posix environment

```
#!/bin/csh
optimize oc disp –anl prob1.anldat –dsgn prob1.dsgndat –opt prob1.ocdat > prob1.output
```

(2) Batch file 2. Alternate C-Shell script for Posix environment

```
#!/bin/csh
optimize oc disp > prob1.output << EOF
prob1.anldat
prob1.dsgndat
prob1.ocdat
EOF
```

(3) Batch file 3. REXX executive for VM/CMS environment

```
/* */
optimize oc disp –anl prob1 –dsgn prob1 –opt prob1 ocdat > prob1 output a
```

(4) Batch file 4. Alternate REXX executive for VM/CMS environment

```
/* */
makebuf
queue ‘prob1 anldat a’
queue ‘prob1 dsgndat a’
queue ‘prob1 ocdat a’
optimize oc disp > prob1 output a
dropbuf
```

Section 5
Illustrative Examples

Three examples are given in this section to illustrate execution with CometBoards. Included with each example is a brief description of the problem, a listing of the user-supplied data files, and listing of the output produced by CometBoards or tables depicting the results. These examples were run on an Iris workstation.

5.1 Example 1: Three-Bar Truss

The three bar truss, shown in figure 3, is taken as the first example. The truss is made of a material with Young's modulus *E* of 30 000 ksi, Poisson's ratio ν of 0.3, and weight density ρ of 0.284 lb/in.^{3} The load is at node 1, 50 kips in the *X* direction and 100 kips in the *Y* direction. Stress limits are the only behavior constraints. The stress allowables are tension, 20.0 ksi; compression, 20.0 ksi; and shear, 20.0 ksi. For this example the default values set in CometBoards were used and only the required data input is given. The SUMT optimization technique was used, with the displacement analyzer. The three required data files for running this problem are named:

bar3.anldat
bar3.dsgndat
bar3.sumtdat

The contents of each of these input data files, as well as the output produced from the run, are shown on the following pages.

To run the problem, CometBoards executes the C-shell script file entitled "optimize." This file loads the analyzer, optimizer, and data files and executes the program. To invoke, type

optimize sumt disp > bar3.sumtout

After the prompt "Enter the name of the anldat file," type

bar3.anldat

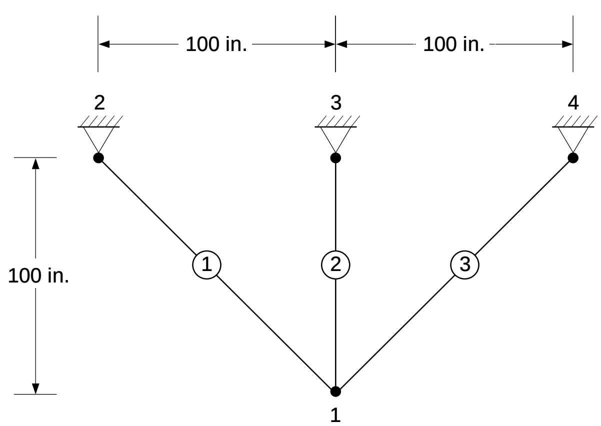


Figure 3.—Optimum design of three-bar truss. (Elements are circled; nodes are not.)

After the prompt “Enter the name of the dsgndat file,” type

bar3.dsgndat

After the prompt “Enter the name of the optimization parameter file,” type

bar3.sumtdat

The problem will run and the output produced will be stored in the bar3.sumtout file. The three input data files required to run this problem and a listing of this output follows.

A C-Shell script file may also be written in order to run this problem in batch mode. A sample of this is given in section 4.

bar3.anldat

```
TITLE:
Three Bar Truss
$
I DIMENSION = 2
$
CONNECTIVITY:
   2     1     2
   2     1     3
   2     1     4
   0     0
$
COORDINATES IN INCHES!
COORDINATES:
    0.000     0.000
 -100.000   100.000
    0.000   100.000
  100.000   100.000
   -1.E75    -1.E75
$
NODAL RESTRAINTS:
    2    2    1    2
    3    2    1    2
    4    2    1    2
    0    0
$
LOADS IN KIPS!
LOAD CONDITIONS:
    1    2    1    50.000    2    100.000
    0    0
    0    0
```

bar3.dsgndat

```
STRESS IN KSI!
STRENGTH:
20.00  20.00  20.00
```

bar3.sumtdat

```
$
```

bar3.sumtout

```
To solve a problem defined in the following data files:
  bar3.anldat
  bar3.dsgndat
  bar3.sumtdat
using this optimizer:  A Sequence of Unconstrained Minimizations (NEWSUMT)
and this analyzer:  The Displacement Method


1
```

```
 NNNNNNN    NNNN      AAAAAA      SSSSSSSSSSS       AAAAAA
 N.....NN   N..N      A.....A     S.........S       A.....A
 N..NN..N   N..N     A..AA..A    S..........S      A..AA..A
 N..NN..N   N..N     A..AA..A    S..SSSSSSSSS      A..AA..A
 N..NN..N   N..N     A..A A..A   S..S              A..A A..A
 N..N N..N  N..N    A..A  A..A   S..SSSSSSSS      A..A  A..A
 N..N N..N  N..N    A..A  A...A  SS.........SS    A..A  A...A
 N..N N...N N..N   A...A   A..A   SSSSSSSS...S   A...A   A..A
 N..N  N..N N..N   A..A    A..A          S..S   A..A    A..A
 N..N  N..N N..N  A...A     A..A        SS..S  A...A     A..A
 N..N   N.NNN..N  A..A      A..A  SSSSSSSS..S   A..A      A..A
 N..N   N......N  A..A      A..A  S.........S   A..A      A..A
 NNNN   NNNNNNNN AAAA       AAAA SSSSSSSSSS   AAAA       AAAA


  DDDDDDD   EEEEEEEE   SSSSSSSS  II    GGGGGG   NNN      NN
  DDDDDDDD  EEEEEEEE  SSSSSSSSS  II   GGGGGGGG  NNNN     NN
  DD    DDD EE        SS         II  GG         NN NN    NN
  DD     DD EEEEEE     SSSSSSS   II  GG   GGGG  NN  NN   NN
  DD     DD EE               SS  II  GG     GG  NN   NN  NN
  DD    DDD EE               SS  II  GG     GG  NN    NN NN
  DDDDDDDD  EEEEEEEEE SSSSSSSSS  II   GGGGGGGG  NN     NNNN
  DDDDDDD   EEEEEEEEE SSSSSSSS   II    GGGGGG   NN      NNN


               **********************
               *  CASE 6:  NEWSUMT  *
               **********************

1


      ***********************************
       ***    Echo of Analysis data    ***
      ***********************************


 TITLE:
 Three Bar Truss
 $
 I DIMENSION = 2
 $
 CONNECTIVITY:
     2    1    2
     2    1    3
     2    1    4
     0
 $
 COORDINATES IN INCHES!
 COORDINATES:
       0.00000      0.00000
    -100.00000    100.00000
       0.00000    100.00000
     100.00000    100.00000
 **************************
 $
 NODAL RESTRAINTS:
     2    2    1    2
     3    2    1    2
     4    2    1    2
     0    0
 $
 LOADS IN KIPS!
 LOAD CONDITIONS:
    1    2    1     50.00000    2    100.00000
    0    0
    0    0
1


         ***********************************
          *** Analyzer parameters read in ***
         ***********************************



 TITLE:  Three Bar Truss

 DIMENSION  =     2
 ELEMENTS   =     3            NODES =     4     BOUNDARIES =     6
 LOAD COND  =     1     MATERIAL PROP =     1     PRINT CTRL =     0

 ELEMENT CONNECTIVITIES:
 ELEMENT    TYPE   ELEM-1   ELEM-2   ELEM-3   ELEM-4
 -------    ----   ------   ------   ------   ------
    1        2        1        2        0        0
    2        2        1        3        0        0
    3        2        1        4        0        0
```

```
 COORDINATES:
 Coordinate units are in inches!
 NODE         X             Y             Z
 ----     ----------    ----------    ----------
    1       0.00000       0.00000       0.00000
    2    -100.00000     100.00000       0.00000
    3       0.00000     100.00000       0.00000
    4     100.00000     100.00000       0.00000

 MATERIAL PROPERTIES:
 Young's modulus in ksi!
 Density in lbs/cubic in.!
 MATERIAL TYPE    YOUNG'S MODULUS    POISSON'S RATIO       DENSITY
 -------------    ---------------    ---------------       -------
        1           0.30000E+05          0.30000           0.28400

 NODAL RESTRAINTS:
 NODE   DIRECTION
 ----   ---------

    2       1
    2       2
    3       1
    3       2
    4       1
    4       2

 LOAD CONDITION:    1
 Loads given in kips converted to pounds!
 NODE         X             Y             Z
 ----     ----------    ----------    ----------
    1    50000.00000   100000.00000
1
      ***********************************
       ***    Echo of  DESIGN  data    ***
      ***********************************



 STRESS IN KSI!
 STRENGTH:
     0.20000E+02    0.20000E+02    0.20000E+02
1


         ***********************************
          ***  Design parameters read in  ***
         ***********************************



 NUMBER OF STRESS CONSTRAINTS PER LOAD CONDITION =     3
 Stress units given in ksi converted to psi!
                                        STRESS LIMITATIONS
                               --------------------------------------
 CONSTRAINT    ELEMENT         TENSION   COMPRESSION        SHEAR
 ----------    -------      -----------  -----------  -----------
      1           1         0.20000E+05  0.20000E+05  0.20000E+05
      2           2         0.20000E+05  0.20000E+05  0.20000E+05
      3           3         0.20000E+05  0.20000E+05  0.20000E+05

 NUMBER OF DISPLACEMENT CONSTRAINTS PER LOAD CONDITION =     0
1

 DESIGN VARIABLE    MINIMUM BOUNDS   INITIAL DESIGN    UPPER BOUNDS
 ---------------    --------------   --------------    ------------
      1               0.01000           1.00000         .10000E+05
      2               0.01000           1.00000         .10000E+05
      3               0.01000           1.00000         .10000E+05
1



1
       ********************************
        ***   Echo of NEWSUMT data   ***
       ********************************


 $
1
```

```
     ************************************
    ***  NEWSUMT parameters read in  ***
   ************************************

 NDV and NCON are      3,      3
 LOBJ is     1
 The ILIN array is
          0    0    0
 The ISIDE array is
          3    3    3
 IFD is     0
 JPRINT is     0
 EPSGSN, EPSODM, and EPSRSF are   1.000000E-03,   1.000000E-03,   5.000000E-03
 RA, RACUT, RAMIN, and MFLAG are   1.00000,   0.10000,   0.00000,    0
 STEPMX is   2.00000
 MAXGSN, MAXODM, and MAXRSF are      20,       6,      15
1********************************************  N E W S U M T   O P T I M I Z E R ********************************************

                                              CONTROL PARAMETERS

                          INITIAL TRANSITION POINT . . . . . . . . . G0 =  1.0000E-01
                          TRANSITION POINT EXPONENT  . . . . . . . .  P =  5.0000E-01
                          INITIAL TRANSITION POINT COEFFICIENT . . .  C =  2.0000E-01
                          GOLDEN SECTION CONVERGENCE . . . . . . EPSGSN =  1.0000E-03
                          UNCONSTRAINED MINIMIZATION CONVERGENCE EPSODM =  1.0000E-03
                          CONVERGENCE AMONG RESPONSE SURFACES  . EPSRSF =  5.0000E-03
                          RESPONSE FACTOR REDUCTION  RATIO . . .  RACUT =  1.0000E-01
                          MINIMUM ALLOWABLE RESPONSE FACTOR  . . IRAMIN =  1.0000E-13
                          MAXIMUM ALLOWABLE STEP SIZE  . . . . . STEPMX =  2.0000E+00
                          MAXIMUM ALLOWABLE GOLDEN SECTIONS  . . MAXGSN =   20
                          MAXIMUM NUMBER OF ODM PER SURFACE  . . MAXODM =    6
                          MAXIMUM ALLOWABLE RESPONSE SURFACES  . MAXRSF =   15
                          PRINTOUT CONTROL . . . . . . . . . . . JPRINT =    0
                          FINITE DIFFERENCE GRADIENT CONTROL . . .  IFD =    0
                          APPROXIMATION CONTROL FLAG . . . . . . IFLAPP =    0

                                              SYSTEM PARAMETERS

                          NUMBER OF DESIGN VARIABLES . . . . . . .  NDV =    3
                          NUMBER OF EFFECTIVE CONSTRAINTS  . . . . NTCE =    3


    INITIAL DESIGN ANALYSIS SUMMARY
    -------------------------------
         INITIAL DESIGN VARIABLE VECTOR
           1.0000E+00  1.0000E+00  1.0000E+00
         SIDE CONSTRAINTS
               -1          1         -2          2         -3          3
          9.9000E-01  9.9990E+03  9.9000E-01  9.9990E+03  9.9000E-01  9.9990E+03
         CONSTRAINTS
          6.9670E-01 -1.9289E+00 -2.2322E+00
         OBJECTIVE FUNCTION =  1.0872733E+02
1-----------------------------------------

  OPTIMIZATION OF RESPONSE SURFACE NO.  1    PENALTY MULTIPLIER =   2.434588E+01

 -----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 3.317006E+04  OBJECTIVE FUNCTION= 2.206714E+02  MOVE DISTANCE= 1.801083E+00
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 1.455932E+03  OBJECTIVE FUNCTION= 3.741090E+02  MOVE DISTANCE= 2.420157E+00
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 1.132875E+03  OBJECTIVE FUNCTION= 2.452627E+02  MOVE DISTANCE= 3.358192E+00
    ONE DIMENSIONAL SEARCH 4  TOTAL FUNCTION= 5.795974E+02  OBJECTIVE FUNCTION= 3.395172E+02  MOVE DISTANCE= 1.689693E+00
    ONE DIMENSIONAL SEARCH 5  TOTAL FUNCTION= 5.647102E+02  OBJECTIVE FUNCTION= 3.313338E+02  MOVE DISTANCE= 6.961569E-01
    ONE DIMENSIONAL SEARCH 6  TOTAL FUNCTION= 5.625562E+02  OBJECTIVE FUNCTION= 3.465384E+02  MOVE DISTANCE= 2.815084E-01
1-----------------------------------------

  OPTIMIZATION OF RESPONSE SURFACE NO.  2    PENALTY MULTIPLIER =   2.434588E+00

 -----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 3.090257E+02  OBJECTIVE FUNCTION= 2.624901E+02  MOVE DISTANCE= 1.477614E+00
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 3.055673E+02  OBJECTIVE FUNCTION= 2.546706E+02  MOVE DISTANCE= 1.991956E-01
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 3.055171E+02  OBJECTIVE FUNCTION= 2.549733E+02  MOVE DISTANCE= 2.931387E-02
    ONE DIMENSIONAL SEARCH 4  TOTAL FUNCTION= 3.055167E+02  OBJECTIVE FUNCTION= 2.551025E+02  MOVE DISTANCE= 2.454380E-03
1-----------------------------------------

  OPTIMIZATION OF RESPONSE SURFACE NO.  3    PENALTY MULTIPLIER =   2.434588E-01

 -----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 2.408495E+02  OBJECTIVE FUNCTION= 2.272096E+02  MOVE DISTANCE= 4.806601E-01
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 2.406277E+02  OBJECTIVE FUNCTION= 2.264800E+02  MOVE DISTANCE= 2.312932E-02
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 2.406241E+02  OBJECTIVE FUNCTION= 2.265644E+02  MOVE DISTANCE= 4.263435E-03
1-----------------------------------------

  OPTIMIZATION OF RESPONSE SURFACE NO.  4    PENALTY MULTIPLIER =   2.434588E-02

 -----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 2.218282E+02  OBJECTIVE FUNCTION= 2.176429E+02  MOVE DISTANCE= 1.513454E-01
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 2.218172E+02  OBJECTIVE FUNCTION= 2.175513E+02  MOVE DISTANCE= 3.198395E-03
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 2.218171E+02  OBJECTIVE FUNCTION= 2.175617E+02  MOVE DISTANCE= 2.178294E-04
1-----------------------------------------
```

```
  OPTIMIZATION OF RESPONSE SURFACE NO.  5     PENALTY MULTIPLIER =   2.434588E-03

 ----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 2.160440E+02  OBJECTIVE FUNCTION= 2.147135E+02  MOVE DISTANCE= 4.805111E-02
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 2.160435E+02  OBJECTIVE FUNCTION= 2.147177E+02  MOVE DISTANCE= 4.721359E-04
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 2.160435E+02  OBJECTIVE FUNCTION= 2.147181E+02  MOVE DISTANCE= 4.257247E-05
1----------------------------------------

  OPTIMIZATION OF RESPONSE SURFACE NO.  6    PENALTY MULTIPLIER =   2.434588E-04

 ----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 2.142491E+02  OBJECTIVE FUNCTION= 2.139016E+02  MOVE DISTANCE= 1.373370E-02
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 2.142359E+02  OBJECTIVE FUNCTION= 2.138364E+02  MOVE DISTANCE= 1.086851E-03
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 2.142352E+02  OBJECTIVE FUNCTION= 2.138220E+02  MOVE DISTANCE= 2.415765E-04
1----------------------------------------

  OPTIMIZATION OF RESPONSE SURFACE NO.  7    PENALTY MULTIPLIER =   2.434588E-05

 ----------------------------------------
    ONE DIMENSIONAL SEARCH 1  TOTAL FUNCTION= 2.136701E+02  OBJECTIVE FUNCTION= 2.135619E+02  MOVE DISTANCE= 4.372896E-03
    ONE DIMENSIONAL SEARCH 2  TOTAL FUNCTION= 2.136653E+02  OBJECTIVE FUNCTION= 2.135391E+02  MOVE DISTANCE= 3.807024E-04
    ONE DIMENSIONAL SEARCH 3  TOTAL FUNCTION= 2.136650E+02  OBJECTIVE FUNCTION= 2.135346E+02  MOVE DISTANCE= 7.639320E-05

    FINAL RESULTS OF OPTIMIZATION
    -----------------------------
         CURRENT DESIGN VARIABLE VECTOR
           1.0786E-02  2.5015E+00  3.5370E+00
         SIDE CONSTRAINTS
                -1          1         -2          2         -3          3
          7.8640E-04  1.0000E+04  2.4915E+00  9.9975E+03  3.5270E+00  9.9965E+03
         CONSTRAINTS
           9.9983E-01  5.9108E-04  4.1804E-04
         TOTAL FUNCTION     =  2.1366503E+02
         OBJECTIVE FUNCTION =  2.1353460E+02

         FINAL STATISTICS                                                      CUMULATIVE CPU(SEC)
         ----------------
              NUMBER OF RESPONSE SURFACE . . . .   7                  TOTAL                0.0000
              NUMBER OF ONE DIMENSIONAL SEARCH .  25                  O.D.M.               0.0000
              NUMBER OF ANALYSES                                      DIRECTION            0.0000
                                                                      EQ. SOLVER           0.0000
                   OBJECTIVE FUNCTION . . . . . . . . . . . . .  259                       0.0000
                   GRADIENT OF OBJECTIVE FUNCTION . . . . . . .    1                       0.0000
                   CONSTRAINT FUNCTIONS . . . . . . . . . . . .   51                       0.0000
                   GRADIENT OF LINEAR CONSTRAINT FUNCTIONS  . .    0                       0.0000
                   GRADIENT OF NONLINEAR CONSTRAINT FUNCTIONS .   25                       0.0000
                   APPROXIMATE CONSTRAINT FUNCTIONS . . . . . .  208                       0.0000
1

   +*********************************************+
    |                                            |
    |          NEWSUMT  Final Results.           |
    |                                            |
   +*********************************************+

    This solves a Three Bar Truss problem with the following
    number of constraint types:

    Stress constraints      =    3


 The objective function at the optimal design:   2.13534598E+02

 The total execution time was:          0.830 seconds.


1       Design      Lower      Initial      Upper          Optimum
       Variable     Bound      Design       Bound          Design
       --------     -----      -------      -----          -------

           1      0.01000     1.00000    1.000E+04        0.01079
           2      0.01000     1.00000    1.000E+04        2.50148
           3      0.01000     1.00000    1.000E+04        3.53701


 Optimal Design is Feasible!
1           Constraint      Active      Constraint     Passive
             Number      Constraint      Value       Constraint
            ----------    ----------    ----------    ----------

                1                       -0.99983      PASSIVE
                2          ACTIVE       -0.00059
                3          ACTIVE       -0.00042

Active Stress Constraints = 2
```

5.2 Example 2: Ten-Bar Truss With Linking

The 10-bar truss, shown in figure 4, was taken as the second example. This example was obtained by linking the variables into a set of five independent linked design groups as shown in table 11. This truss is made of aluminum, with Young's modulus E of 10 000 ksi, Poisson's ratio ν of 0.3, and weight density ρ of 0.1 lb/in.^{3} The truss is subjected to one load condition and has two displacement constraints and one frequency constraint.

This problem was run on an Iris workstation by using the displacement analyzer with (1) the FUD optimization technique and (2) all 16 OC methods. The contents of the data files are shown in the following pages. The results are depicted in table 12.

(1) To run CometBoards using FUD, type

optimize fud disp > bar10.fudout

and enter the following data file names at the prompts

bar10.anldat
bar10.dsgndat
bar10.fuddat

followed by an **n** to answer the prompt for interactive input. The results will be stored in the bar10.fudout file.

(2) The files required to run OC are named

bar10.anldat
bar10.dsgndat
bar10.ocdat1 through **bar10.ocdat16**

To run the first OC method, for example, type

optimize oc disp –anl bar10.anldat –dsgn bar10.dsgndat –opt bar10.ocdat1 > bar10.ocout1

and answer the prompt for interactive input with an **n**. Note that all methods use the same analysis and design data files, anldat and dsgndat, and only the optimization data files differ.

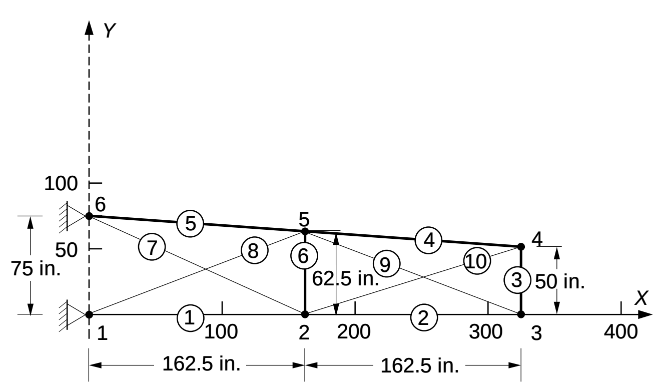


Figure 4.—Optimum design of 10-bar truss. (Elements are circled; nodes are not.)

TABLE 11.—DESIGN VARIABLE LINKING OF 10-BAR TRUSS

Serial number	Design variable	Members linked
1	1	1
2	2	2,4
3	3	5
4	4	3,9,10
5	5	6,7,8

TABLE 12.—RESULTS FOR 10-BAR TRUSS

Methods	OC methods		Optimum weight, lb	Optimum design variables (areas in square inches)				
	L	X		A1	A2	A3	A4	A5
1	1	1	444.351	6.851	1.945	6.591	1.771	2.225
2	1	2	444.260	6.853	1.940	6.604	1.766	2.227
3	1	3	444.112	6.857	1.930	6.618	1.758	2.231
4	2	1	452.938	6.758	2.245	6.292	2.019	2.116
5	2	2	454.974	7.177	1.752	7.206	1.620	2.407
6	2	3	443.173	6.918	1.782	6.903	1.638	2.302
7	3	1	1263.502	19.309	6.261	17.377	5.633	5.902
8	3	2	1282.120	19.309	6.354	17.633	5.716	5.989
9	3	3	1641.241	24.717	8.133	22.572	7.317	7.666
10	4	1	2035.628	30.657	19.309	27.996	9.076	9.508
11	4	2	5853.529	88.155	29.008	80.504	26.097	27.342
12	4	3	756.686	5.180	5.180	5.180	5.180	5.180
13	5	4	460.207	6.950	2.282	6.346	2.029	2.156
14	5	5	2035.628	30.657	10.088	27.996	9.076	9.508
15	5	6	5853.529	88.155	29.008	80.504	26.097	27.342
16	5	7	756.686	5.180	5.180	5.180	5.180	5.180
FUD			756.686	5.180	5.180	5.180	5.180	5.180

bar10.anldat

```
TITLE:
Ten Bar Truss    (ANDLAT/bar10)
$
I DIMENSION = 2
$
MATERIAL PROPERTIES:
 10000.0    0.300    0.100
     0.0    0.       0.
$
CONNECTIVITY:
   2     1     2
   2     2     3
   2     3     4
   2     4     5
   2     5     6
   2     2     5
   2     2     6
   2     1     5
   2     3     5
   2     2     4
   0     0
$
COORDINATES IN INCHES!
COORDINATES:
    0.000     0.000
  162.500     0.000
  325.000     0.000
  325.000    50.000
  162.500    62.500
    0.000    75.000
   -1.E75    -1.E75
$
NODAL RESTRAINTS:
    1    2    1    2
    6    2    1    2
    0    0
$
LOADS IN KIPS!
LOAD CONDITIONS:
    2    2   1    0.60    2   6.00
    3    2   1    6.00    2   6.00
    4    2   1    1.75    2   1.25
    5    2   1    0.175   2   2.50
    0    0
    0    0
$
BEGIN DYNAMIC ANALYSIS PARMS!
FREQPARM = 0.0001
MASS IN POUNDS-F!
LUMPED ELEMENTAL MASS!
EQUIPMENT LUMPED MASS:
2   75.0
3  135.0
4   75.0
5  135.0
0    0.0
END DYNAMIC ANALYSIS PARMS!
```

bar10.dsgndat

```
$
DISP LIMITS BY NODES:
3   2   1.50
4   2   1.50
0   0   0.00
$
FREQUENCY LIMITS:
1 15.0
0  0.0
$
UNIFORM LINKED DV LOW BOUNDS= 0.1
$
DESIGN VARIABLE LINKING:
1   1
2   2   4
1   5
3   3   9   10
3   6   7    8
0
$
LINKED INITIAL DESIGN:
    4.000     4.000     4.000   4.000   4.000
```

bar10.fuddat

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
```

bar10.ocdat1

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0
```

bar10.ocdat2

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0
```

bar10.ocdat3

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0
```

bar10.ocdat4

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0
```

bar10.ocdat5

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0
```

bar10.ocdat6

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0
```

bar10.ocdat7

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 1.0 0.0 0.0
```

bar10.ocdat8

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 1.0 0.0 0.0
```

bar10.ocdat9

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 1.0 0.0 0.0
```

bar10.ocdat10

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
1.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 1.0 0.0
```

bar10.ocdat11

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 1.0 0.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 1.0 0.0
```

bar10.ocdat12

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 1.0 0.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 1.0 0.0
```

bar10.ocdat13

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 0.0 1.0 0.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 0.0 1.0
```

bar10.ocdat14

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 0.0 0.0 1.0 0.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 0.0 1.0
```

bar10.ocdat15

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 0.0 0.0 0.0 1.0 0.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 0.0 1.0
```

bar10.ocdat16

```
$
MAKE 1 ACTIVE CON!
$
ACTIVE CON NUMBER :
 0 1 0 1 0 2 1 3
$
STRATEGIES :
2 1 0
$
X METHOD WEIGHTS :
0.0 0.0 0.0 0.0 0.0 0.0 1.0 0.0
$
LM METHOD WEIGHTS :
0.0 0.0 0.0 0.0 0.0 1.0
```

5.3 Example 3: Rectangular Plate With Reinforced Hole

The plate under consideration is shown in figure 5 (see also ref. 19). Owing to double symmetry, it is sufficient to treat one quadrant of the finite element mesh in figure 6. Isotropic membrane elements are used to model the plate, whereas the reinforcement is approximated by a series of bar elements. Each element is sized independently. Identical material is used for the plate and the reinforcement, with Young’s modulus E of 10 million psi, Poisson’s ratio ν of 0.25, allowable stress of 25 000.0 psi, and density ρ of 0.1 lb/in.^{3} The plate is subjected to one load condition applied at four nodes. It has 43 stress constraints and two displacement constraints.

This problem was run on an Iris workstation using the displacement analyzer and several optimizers: the FUD, FD, SLP, SQP, and SUMT optimization techniques. The contents of the analysis and design data files are shown on the following pages. The command to execute each method is given below. The results from all these runs are depicted in table 13.

To run CometBoards using the FUD optimization technique, enter

optimize fud disp > plate.fudout

enter the following data file names at the prompts:

plate.anldat
plate.dsgndat
plate.fuddat

and enter an **n** to answer the prompt for interactive input. Results will be stored in the plate.fudout file.

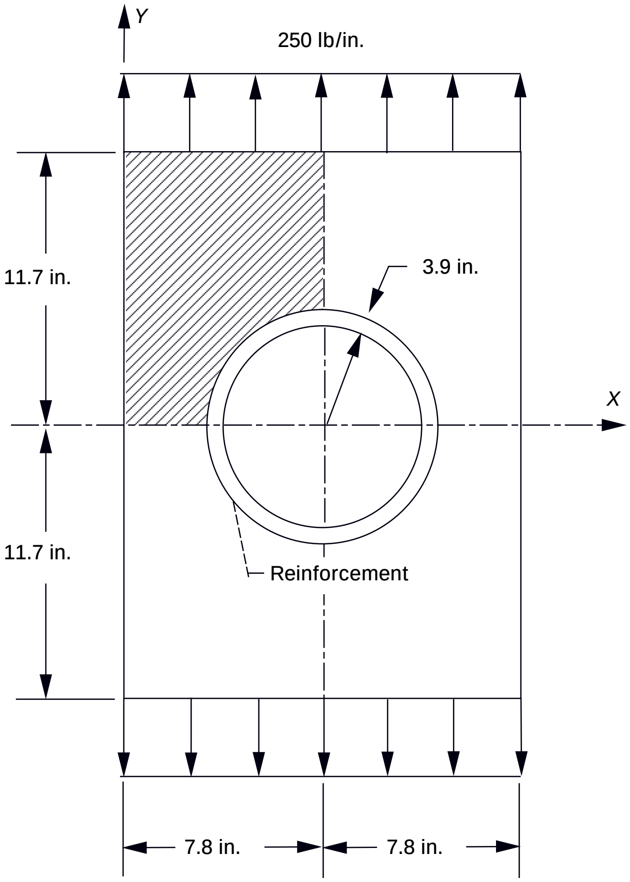
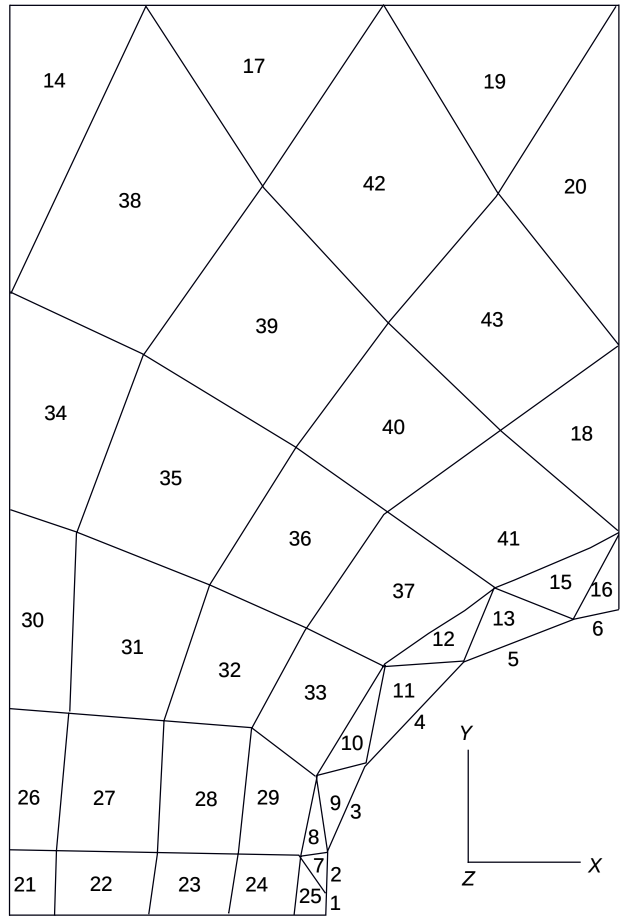


Figure 5.—Plate with reinforced hole.



Fligure 6.—Finite element mesh for quadrant of plate.

TABLE 13.—RESULTS FOR RECTANGULAR PLATE WITH REINFORCED HOLE

Method	Optimum weight, lb
FUD	0.0991
SUMT	.1034
FD	.2078
SLP	.1092
SQP	.0982

To run CometBoards using the FD optimization technique, enter

optimize fd disp > plate.fdout

and enter the following data file names at the prompts:

plate.anldat
plate.dsgndat
plate.fddat

To run CometBoards using the SLP optimization technique, enter

optimize slp disp > plate.slpout

and enter the following data file names at the prompts:

plate.anldat
plate.dsgndat
plate.slpdat

To run CometBoards using the SQP optimization technique, enter

optimize sqp disp > plate.sqpout

and enter the following data file names at the prompts:

plate.anldat
plate.dsgndat
plate.sqpdat

To run CometBoards using the SUMT optimization technique, enter

optimize sumt disp > plate.sumtout

and enter the following data file names at the prompts:

plate.anldat
plate.dsgndat
plate.sumtdat

plate.anldat

```
TITLE:
Rectangular Plate with Reinforced Hole.
I DIMENSION = 2
I PRINT CONTROL = 1
YOUNGS MODULUS IN PSI!
MATERIAL PROPERTIES:
 10000000.0    0.25     0.1
       0.     0.0      0.0
CONNECTIVITY:
    2    6    12
    2   12    13
    2   13    19
    2   19    25
    2   25    31
    2   31    32
    3   11    12    13
    3   11    13    18
    3   13    18    19
    3   18    19    24
```

```
    3   19    24    25
    3   24    25    30
    3   25    30    31
    3   26    33    34
    3   30    31    38
    3   31    32    38
    3   34    35    39
    3   37    38    41
    3   39    40    42
    3   40    41    42
    4    1     2     8     7
    4    2     3     9     8
    4    3     4    10     9
    4    4     5    11    10
    4    5     6    12    11
    4    7     8    15    14
    4    8     9    16    15
    4    9    10    17    16
    4   10    11    18    17
    4   14    15    21    20
    4   15    16    22    21
    4   16    17    23    22
    4   17    18    24    23
    4   20    21    27    26
    4   21    22    28    27
    4   22    23    29    28
    4   23    24    30    29
    4   26    27    35    34
    4   27    28    36    35
    4   28    29    37    36
    4   29    30    38    37
    4   35    36    40    39
    4   36    37    41    40
    0
COORDINATES:
       0.00    0.00
       0.50    0.00
       1.70    0.00
       2.70    0.00
       3.50    0.00
       3.90    0.00
       0.00    0.80
       0.50    0.80
       1.80    0.80
       2.80    0.75
       3.60    0.70
       3.91    0.34
       3.98    0.78
       0.00    2.70
       0.70    2.60
       1.90    2.50
       3.00    2.40
       3.80    1.70
       4.36    1.83
       0.00    5.30
       0.80    5.00
       2.50    4.30
       3.70    3.70
       4.70    3.20
       5.56    3.19
       0.00    8.10
       1.70    7.30
       3.60    6.10
       4.70    5.20
       6.10    4.20
       7.02    3.82
       7.80    3.90
       0.00   11.70
       1.80   11.70
       3.20    9.40
       4.80    7.70
       6.20    6.30
       7.80    4.90
       4.80   11.70
       6.20    9.30
       7.80    7.40
       7.80   11.70
     -1.E75  -1.E75
NODAL RESTRAINTS:
    1    1    2
    2    1    2
    3    1    2
    4    1    2
    5    1    2
    6    1    2
   32    1    1
   38    1    1
   41    1    1
   42    1    1
    0    0
LOADS IN POUNDS!
```

```
LOAD CONDITIONS:
   33    1    2   0.225E+03
   34    1    2   0.600E+03
   39    1    2   0.750E+03
   42    1    2   0.375E+03
    0    0
    0    0
```

plate.dsgndat

```
STRESS IN PSI!
STRESS LIMITS:
25000.0  25000.0  25000.0
UNIFORM DV LOWER BOUNDS= 0.01
DISP LIMITS BY NODES:
     6    1   0.341
    32    2   0.341
     0    0   0.0
 GIVID:
    0.02145  0.02145  0.02145  0.02145  0.02145  0.02145 0.10550  0.10550
    0.10550  0.10550  0.10550  0.10550  0.10550  0.10550 0.10550  0.10550
    0.10550  0.10550  0.10550  0.10550  0.10550  0.10550 0.10550  0.10550
    0.10550  0.10550  0.10550  0.10550  0.10550  0.10550 0.10550  0.10550
    0.10550  0.10550  0.10550  0.10550  0.10550  0.10550 0.10550  0.10550
    0.10550  0.10550  0.10550
```

Section 6
Demonstration Problems

6.1 Introduction

The CometBoards demonstration package currently contains 10 problems. In a single command it automatically loads and executes one of the seven optimization techniques, along with one of the analyzers, on one of the 10 structural problems with as many as three different types of constraints. All the data files and C-shell scripts required to run the demonstration are provided in the demonstration package.

To invoke the demonstration, type the following command:

optimize *optimizer analyzer* **–demo** *problem constraint(s)* > *filename*

where *optimizer* is

sumt	to use the sequence of unconstrained minimizations technique
slp	to use the sequential linear programming technique[1]
fd	to use the modified method of feasible directions[1]
imsl	to use the IMSL version of the successive quadratic programming technique[1]
sqp	to use the sequential quadratic programming technique of IDESIGN
oc	to use the optimality criteria methodology
fud	to use the fully utilized design capability

analyzer is

disp	to use the displacement analyzer
ifm	to use the integrated force method analyzer
sifms	to use the integrated force method analyzer with simplified stress gradients
sifmd	to use the integrated force method analyzer with simplified displacement gradients
sifmsd	to use the integrated force method analyzer with simplified stress and displacement gradients
cfa	to use the closed-form analyzer (available only for three-bar truss example)

problem is

3link	for a three-bar truss with linking and displacement constraints only
3	for a three-bar truss with stress, displacement, and/or frequency constraints
5sub1	for a five-bar truss (top loaded) with displacement constraints only
5sub2	for a five-bar truss (bottom loaded) with displacement constraints only
5sub3	for a five-bar truss (stretch loaded) with displacement constraints only
5sub4	for a five-bar truss (sway loaded) with displacement constraints only
5	for a five-bar truss with stress, displacement, and/or frequency constraints
10	for a 10-bar truss with stress, displacement, and/or frequency constraints
ring	for a 60-bar trussed ring with 24 triangular elements and stress, displacement, and/or frequency constraints
fsw	for a 135-member, forward-swept wing with stress and/or displacement constraints

[1]The SLP, FD, and IMSL optimizers are not available at all sites.

constraint(s) consist of the following constraint types (alone or in any combination):

stress to include stress constraints
disp to include displacement constraints
freq to include frequency constraints

filename is the file name (or on VM/CMS the file identification: filename filetype filemode) for the output file. (The default is for output to go to the screen, if the symbol > and the file name are missing.)

Note that no space is allowed after the hyphen in –demo. Note also that the option flags –wt, –t, and –v as discussed in section 4 may also be used with the demonstration problems. The 10 problems contained in the demonstration are bar truss and membrane examples ranging from 3 to 135 members with stress, displacement, and frequency constraints simultaneously or one at a time. Each of these problems is described in section 6.2.

Any analyzer can be used to run these problems with the exception of CFA, which must be used with the three-bar truss example only. Also, note that frequency constraints are not allowed with the force analyzer or with any of the simplifications of the force analyzer. The commands for running several of the example problems with the available analyzers, optimizers, and constraint types are given below for a Posix environment. To run in the VM/CMS environment, the only change required is the output file name (i.e., filename filetype filemode must be specified instead).

(1) To run the three-bar truss using the FUD technique, the closed-form analyzer, and stress constraints only, type

optimize fud cfa –demo 3 stress

The output will be displayed on the screen. To store the output in a file, type

optimize fud cfa –demo 3 stress > bar3.fudout

(2) To run the three-bar truss problem with linking using the SUMT optimizer, the displacement analyzer with stress, and displacement and frequency constraints, type

optimize sumt disp –demo 3link stress disp freq > bar3link.sumtout

(3) To run the five-bar truss problem using the OC optimizer, the displacement analyzer, and displacement and frequency constraints, type

optimize oc disp –demo 5 disp freq > bar5.ocout

(4) To run the 10-bar truss with the SQP optimizer, the integrated force method analyzer, and displacement constraints only, type

optimize sqp ifm –demo 10 disp > bar10.sqpout

(5) To run the 10-bar truss with the FD technique, the SIFMS analyzer, and stress constraints only, retaining the weight-versus-reanalysis-cycle output (in bar10.wt_iter) along with the weight-versus-CPU-time output (in bar10.wt_time), type

optimize fd sifms –demo 10 stress –wt bar10.wt_iter –t bar10.wt_time > bar10.fdout

(6) To run the fsw problem with the IMSL optimizer and the SIFMSD analyzer with stress and displacement constraints, type

optimize imsl sifmsd –demo fsw stress disp > fsw.imslout

A brief on-line help is available for the demonstration. Type **optimize help** for this information.

6.2 Brief Description of Demonstration Problems

6.2.1 Problem 1: 3link

The three-bar truss shown in figure 3 (but with different node numbering) is included in the CometBoards demonstration package. The three element areas of the structure are linked into two groups each of which is considered as a design variable, as shown in the output file. The three-bar truss has a load of 100.0 kips at node 4, in both *X* and *Y* directions. Nodal restraints are at nodes 1, 2, and 3 in both *X* and *Y* directions. Two different sets of material properties are given. This problem is optimized by using displacement constraints only, where the displacement limits are 0.70710 in. applied at node 4 in both *X* and *Y* directions.

6.2.2 Problem 2: 3

The three-bar aluminum truss mentioned above, but with no linking of design variables, is also included in the demonstration package. The material properties are Young's modulus *E* of 30 000 ksi, Poisson's ratio ν of 0.3, and density ρ of 0.1 lb/in.^{3} It has loads at node 1 of 50.0 kips in the *X* direction and 100.0 kips in the *Y* direction. This problem is optimized by using stress, displacement, and frequency constraints. The stress limit is 20.0 ksi; the displacement limits are at node 1, 0.20 in. in the *X* direction and 0.05 in. in the *Y* direction. The frequency limit is 14.142 Hz. The initial design is 1.0 in.^{2} and the lower bound is 0.001 in.^{2}

6.2.3 Problem 3: 5sub1

The five-bar aluminum truss, shown in figure 7, is included in the demonstration package. The material properties are Young's modulus *E* of 10 000 ksi, Poisson's ratio ν of 0.3, and density ρ of 0.1 lb/in.^{3} It has a load of 100.0 kips at node 4 in the *Y* direction. This problem is optimized by using displacement constraints only. The displacement limit is 2.0 in. at node 4 in the *Y* direction. The initial design is 1.0 in.^{2} and the lower bound is 0.001 in.^{2}

6.2.4 Problem 4: 5sub2

The same five-bar aluminum truss as in problem 3 (fig. 7) but with different loads and nodal restraints is included in the demonstration package. Node 3 is restrained in both the *X* and *Y* directions. The material properties are as before. It has a load of 100.0 kips at node 2 in the *Y* direction. This problem is optimized by using displacement constraints only. The displacement limit is 2.0 in. at node 4 in the *Y* direction. The initial design is 1.0 in.^{2} and the lower bound is 0.001 in.^{2}

6.2.5 Problem 5: 5sub3

The same five-bar aluminum truss is included again with the same material properties and the same nodal restraints as in problem 4 (fig. 7). It has loads of –100.0 kips at node 2 in the *Y* direction and 100.0 kips at node 4 in the *Y* direction. This problem is optimized by using displacement constraints only. The displacement limit is 0.1 in. at nodes 2 and 4 in the *Y* direction. The initial design is 1.0 in.^{2} and the lower bound is 0.001 in.^{2}

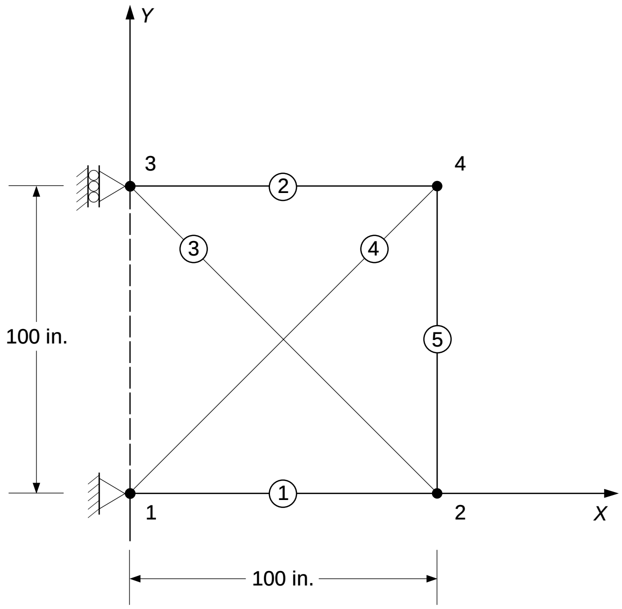


Figure 7.—Optimum design of five-bar truss. (Elements are circled; nodes are not.)

6.2.6 Problem 6: 5sub4

The five-bar aluminum truss is included once more with node 3 fully restrained (fig. 7). The material properties are as before. In this case there are loads of 100.0 kips at node 2 in the *Y* direction and 100.0 kips at node 4 in the *Y* direction. This problem is optimized by using displacement constraints only. The displacement limit is 2.0 in. at nodes 2 and 4 in the *Y* direction. The initial design is 1.0 in.^{2} and the lower bound is 0.001 in.^{2}

6.2.7 Problem 7: 5

The five-bar aluminum truss is included in the demonstration package with multiple constraint types (fig. 7). Again node 3 is restrained in both directions. The material properties are Young's modulus E of 10 000 ksi, Poisson's ratio ν of 0.3, and density ρ of 0.1 lb/in.^{3} It has loads of –1000.0 kips at node 2 in the *Y* direction and 1000.0 kips at node 4 in the *Y* direction. This problem is optimized by using stress, displacement, and frequency constraints. The stress limit is 20.0 ksi; the displacement limits are as follows: at node 2, 0.25 in. in the *X* direction and 0.50 in. in the *Y* direction, and at node 4, 0.25 in. in the *X* direction and 0.50 in. in the *Y* direction. The frequency limit is 10.0 Hz. The initial design is 1.0 in.^{2} and the lower bound is 0.001 in.^{2}

6.2.8 Problem 8: 10

The 10-bar truss aluminum truss (fig. 8) is also included in the CometBoards demonstration package. The truss is subjected to loads at nodes 2 and 3 of –100.0 kips in the *Y* direction. The material properties are Young's modulus E of 10 000 ksi, Poisson's ratio ν of 0.3, and density ρ of 0.1 lb/in.^{3} This problem is optimized by using stress, displacement, and frequency constraints. The stress limit is 25.0 ksi; the displacement limit is 2.0 in. in the *Y* direction at nodes 2, 3, 4, and 5. The frequency limit is 8.0 Hz. The initial design is 1.0 in.^{2} and the lower bound is 0.1 in.^{2}

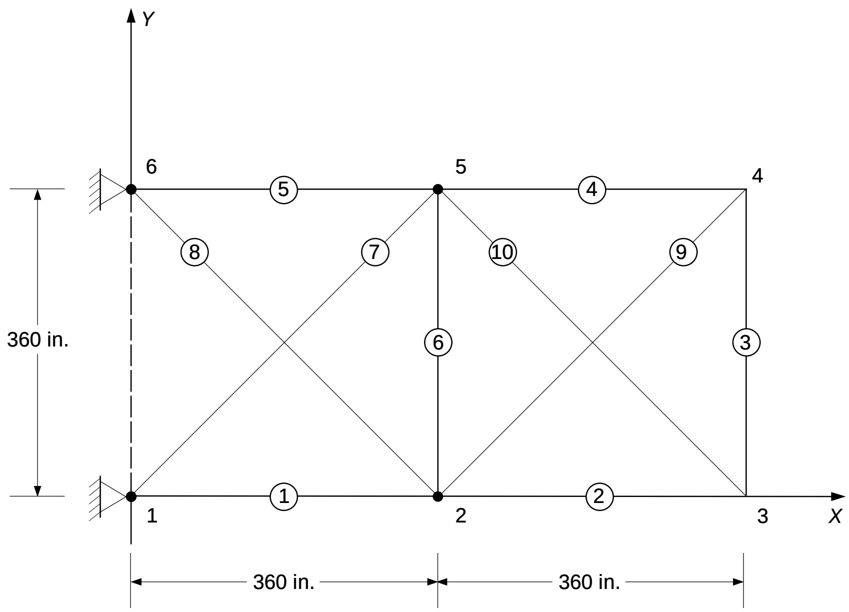


Figure 8.—Optimum design of 10-bar truss. (Elements are circled; nodes are not.)

6.2.9 Problem 9: ring

The two-dimensional, 60-bar trussed ring with 24 triangular elements, shown in figure 9, is included in the CometBoards demonstration package. The 60 element areas of the structure are linked into 25 groups, each of which is considered as a design variable as shown in the output file. The truss is subjected to three load conditions. The material properties are Young's modulus E of 10 000 ksi, Poisson's ratio ν of 0.3, and density ρ of 0.1 lb/in.^{3} This problem is optimized by using stress, displacement, and frequency constraints. The stress limit is 10.0 ksi; the displacement limit is 1.75 in. in the Y direction at node 4. The frequency limit is 13.0 Hz. The initial design is 1.0 in.^{2} and the lower bound is 0.01 in.^{2}

6.2.10 Problem 10: fsw

The finite element model of the forward-swept wing (fsw), shown in figure 10, is included in the CometBoards demonstration package. It has 30 grid points and 135 truss elements. The fsw is made of aluminum with Young's modulus E of 10 000 ksi, Poisson's ratio ν of 0.3, and density ρ of 0.1 lb/in.^{3} This problem is optimized by using stress and displacement constraints, as shown in the output file.

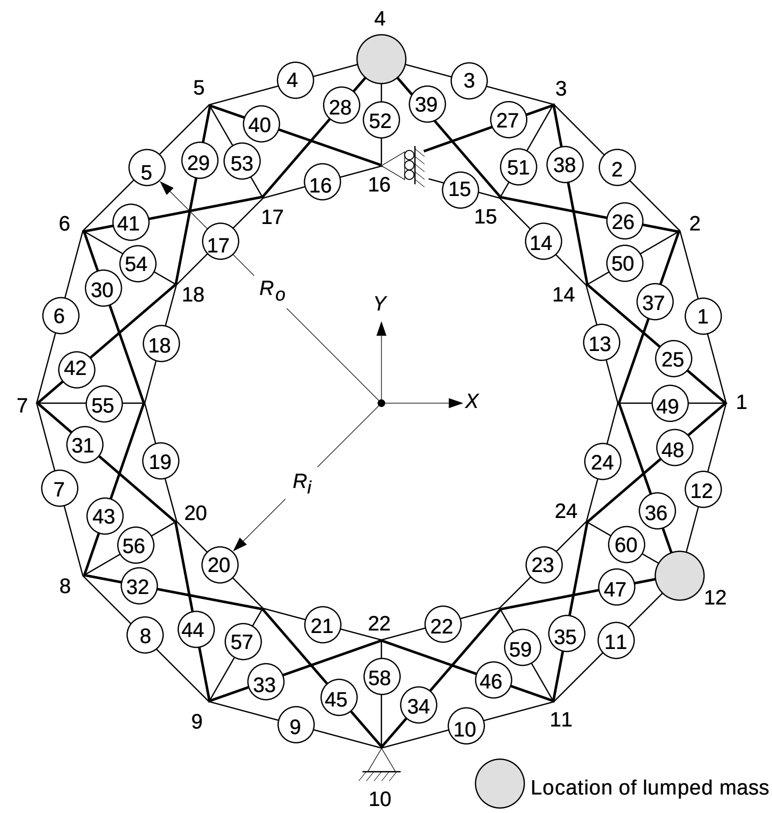


Figure 9.—Optimum design of 60-bar trussed ring. (Elements are circled; nodes are not.)

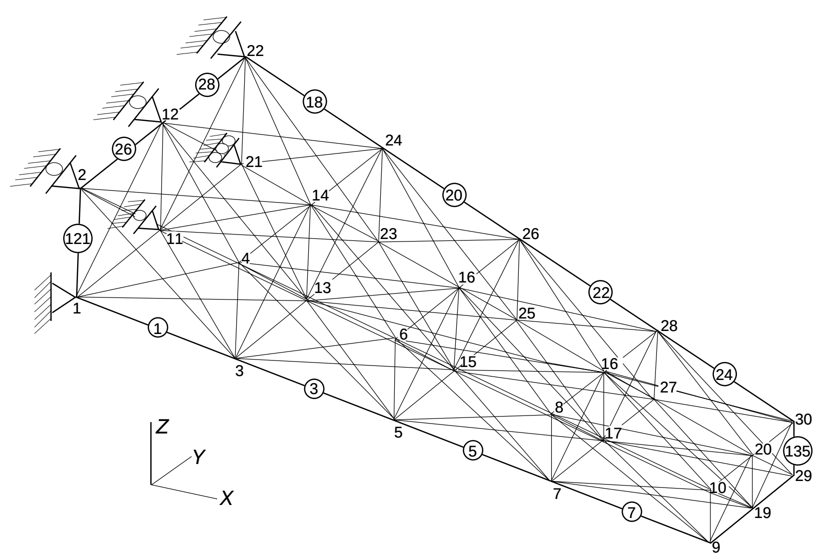


Figure 10.—Optimum design of forward-swept wing. (Representative elements are circled, nodes are not.)

Appendix A

Numerical Experimentation Using Cometboards

To validate the computer code and to examine the performance of different nonlinear mathematical programming algorithms for different problem types, CometBoards was used to solve several numerical examples on a Silicon Graphics Iris/Posix workstation. Optimum solutions for a set of 29 structural design problems solved by using CometBoards are summarized in this appendix for the four optimization methods available at most sites. The examples have been divided into three groups depending on the number of design variables. Problems in group I include examples with 10 to 20 design variables. Problems with 21 to 50 design variables are included under group II. Group III includes problems that have more than 51 design variables.

Optimum designs for multiple load conditions under stress, displacement, and frequency constraints have been obtained for all 29 examples by using four different design methods. The results of 203 runs are summarized in tables 14 to 21 and figures 11 to 13, for the benefit of users of the code CometBoards. The information given in tables 14 to 21 is explained below.

Take problem 16 in table 18(f) for example: The title of table 18(f), “Design of a 60-bar trussed ring for stress, displacement, and frequency constraints (25 linked design variables),” indicates that the structure is a ring made of 60 bar elements. The 60 bars are linked to obtain a nonlinear programming problem with 25 design variables. The example belongs to problem group II. The problem has 60 stress constraints and one displacement constraint for each of the three load conditions and one frequency constraint. As shown in the first column of table 18(f), this problem is number 16 and has a total of 184 behavior constraints: 180S represents 180 stress constraints, 3D represents 3 displacement constraints, and 1F represents 1 frequency constraint. At optimum several behavior constraints are active as indicated in table 18(f) under column 3, “Number of active constraints”: 22S represents 22 active stress constraints, 1D represents 1 active displacement constraint, and 1F represents 1 active frequency constraint using SUMT. The fourth column in table 18(f), “CPU time, sec” provides CPU time in seconds on a Silicon Graphics Iris 4D/35 Posix workstation. The fifth column, “Optimum weight, lb,” provides the weight of the optimum design in pounds. The last column provides grades between A and F:

(1) Grade A represents less than 1 percent error in optimum weight.
(2) Grade B represents 1 to 10 percent error.
(3) Grade C represents 11 to 20 percent error.
(4) Grade D represents 21 to 50 percent error.
(5) Grade F means that the optimizer failed, with error exceeding 50 percent.

The optimization method SQP performed well when solving the 60-bar trussed ring summarized in table 18(f), achieving a grade of A. SQP solved the problem in the shortest time (i.e., in 354.2 CPU seconds on a Silicon Graphics Iris workstation). SUMT achieved B grade and its solution required 1553.2 CPU seconds. FUD failed, as expected, because of the presence of active displacement and frequency constraints.

For this set of 29 problems any design optimization algorithm can achieve at best 29 perfect scores of grade A. The number of A and B grades achieved by the four design methods for the 29 problems were as follows:

(1) SUMT scored 24 A grades and 4 B grades.
(2) SQP scored 23 A grades and 4 B grades.
(3) OC scored 13 A grades and 11 B grades.
(4) FUD scored 8 A grades and 4 B grades.

The SUMT and SQP optimizers outperformed other methods, scoring 24 and 23 A grades, respectively.

The total CPU times required to solve all 29 problems were

(1) 49 858.6 sec for SUMT (Silicon Graphics Iris workstation)
(2) 18 728.0 sec for SQP (Silicon Graphics Iris workstation)

The solutions of the 29 problems showed that SUMT is the most reliable optimizer of the four reported. It is robust but may require excessive computations. The second place goes to SQP, which was found to be the fastest in terms of CPU time.

To examine the relative performance of the methods in reaching an optimum solution, the optimum weights obtained for the 29 problems by the four optimization methods are presented in bar diagrams in figures 11 to 13. For small problems, which belong to group I (10 to 20 design variables), performance was acceptable (fig. 11) for most optimization methods. Most optimizers provided acceptable results (fig. 12) for group II problems (21 to 50 design variables). The exceptions were OC (optimality criteria) and FUD (fully utilized design). The performance for a problem with 57 active design variables is presented in figure 13. For this difficult problem SUMT (sequence of unconstrained minimization techniques) provided the correct optimum design.

TABLE 14.—DESIGN OF 10-BAR TRUSS WITH STRESS, DISPLACEMENT, AND FREQUENCY CONSTRAINTS

(a) Ten bar areas as independent design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
1. 20S,4D,1F	SUMT	7S,2D,1F	7.9	3326.94	A
	SQP	8S,2D,1F	2.9	3326.73	A
	OC	5S,2D,1F	46.9	3563.72	B
	FUD	1F	3.1	3488.72	B

(b) Five linked design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
2. 20S,4D,1F	SUMT	6S,2D,1F	8.5	3572.53	A
	SQP	6S,2D,1F	3.5	3571.12	A
	OC	6S,2D,1F	38.8	3580.95	A
	FUD	6S,2D,1F	0.6	3560.02	A

TABLE 15.—DESIGN OF STIFFENED 10-BAR TRUSS WITH STRESS, DISPLACEMENT, AND FREQUENCY CONSTRAINTS

(a) Eighteen design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
3. 36S,4D,1F	SUMT	8S	14.3	1437.65	A
	SQP	9S	15.2	1441.21	A
	OC	5S	199.8	1457.90	B
	FUD	9S	13.6	1673.74	C

(b) Two linked design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
4. 36S,4D,1F	SUMT	2S	17.0	2566.81	A
	SQP	2S	3.9	2567.72	A
	OC	2S	144.5	2567.72	A
	FUD	2S	5.4	2567.72	A

TABLE 16.—DESIGN OF CANTILEVER MEMBRANE DISCRETIZED IN EIGHT TRIANGULAR ELEMENTS FOR STRESS, DISPLACEMENT, AND FREQUENCY CONSTRAINTS

(a) Eight design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
5. 16S,4D,1F	SUMT	6S	8.3	1438.67	A
	SQP	6S	5.5	1440.24	A
	OC	4S	105.6	1480.73	B
	FUD	7S	6.9	1673.29	C

(b) One linked design variable

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
6. 16S,4D,1F	SUMT	1S	7.9	2565.83	A
	SQP	1S	1.3	2566.10	A
	OC	1S	7.0	2566.75	A
	FUD	1S	0.8	2566.75	A

TABLE 17.—DESIGN OF CANTILEVER MEMBRANE FOR STRESS, DISPLACEMENT, AND FREQUENCY CONSTRAINTS

(a) Discretized in 16 quadrilateral elements; 16 design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
7. 32S,4D,1F	SUMT	1S,4D	425.7	3864.73	A
	SQP	-------	3.4	2031.34	F
	OC	4D	241.1	3865.48	A
	FUD	4D	40.7	3887.60	B

(b) Discretized in 32 quadrilateral elements; one linked design variable

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
8. 64S,4D,1F	SUMT	1S,4D	2 203.4	4263.81	A
	SQP	4D	2 028.9	4265.89	A
	OC	4D	4 441.8	4265.92	A
	FUD	4D	277.4	4279.01	A

(c) Discretized in 48 quadrilateral elements; 16 linked design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
9. 96S,4D,1F	SUMT	13S,4D	4 147.8	4671.13	A
	SQP	13S,4D	3 501.9	4566.92	B
	OC	13S,2D	15 578.6	4849.05	B
	FUD	18S,4D	372.0	4812.22	B

(d) Discretized in 64 quadrilateral elements; 16 linked design variables

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
10. 128S,4D,1F	SUMT	32S,4D	5 268.6	4988.70	A
	SQP	29S,4D	2 740.9	4824.34	B
	OC	23S,2D	21 638.6	5149.29	B
	FUD	31S,4D	474.3	4989.72	A

TABLE 18.—DESIGN OF 60-BAR TRUSSED RING
[25 linked design variables.]

(a) For stress constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
11. 180S	SUMT	38S	194.6	280.61	A
	SQP	35S	45.3	276.45	B
	OC	40S	64.4	280.74	A
	FUD	40S	6.9	280.75	A

(b) For displacement constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
12. 3D	SUMT	1D	293.9	270.33	A
	SQP	1D	125.1	270.15	A
	OC	1D	51.6	270.15	A
	FUD	1D	1.4	390.11	D

(c) For frequency constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
13. 1F	SUMT	-------	952.8	334.50	C
	SQP	1F	685.8	378.20	A
	OC	1F	1275.3	368.00	B
	FUD	-------	10.8	630.47	F

(d) For stress and displacement constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
14. 180S,24D	SUMT	28S,1D	401.7	305.27	A
	SQP	30S,1D	113.9	303.80	A
	OC	18S,1D	1227.5	319.20	B
	FUD	1D	7.9	328.90	B

(e) For displacement and frequency constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
15. 24D,1F	SUMT	1D,1F	815.3	358.08	B
	SQP	1D,1F	774.0	378.03	A
	OC	1D,1F	2895.2	373.10	B
	FUD	——	10.9	630.47	F

(f) For stress, displacement, and frequency constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
16. 180S,3D,1F	SUMT	22S,1D,1F	1553.2	411.55	B
	SQP	21S,1D,1F	354.2	414.60	A
	OC	15S,1D,1F	4445.5	434.00	B
	FUD	-------------	54.6	726.71	F

TABLE 19.—DESIGN OF STIFFENED 60-BAR TRUSSED RING
[49 linked design variables.]

(a) For stress constraints

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
17. 252S	SUMT	75S	303.1	157.23	A
	SQP	75S	577.9	156.95	A
	OC	59S	1019.6	179.54	C
	FUD	59S	85.8	179.54	C

(b) For displacement constraints

18. 3D	SUMT	1D	364.6	136.89	A
	SQP	1D	1182.7	135.43	B
	OC	1D	995.6	136.98	A
	FUD	1D	3.9	179.42	D

(c) For frequency constraints

19. 1F	SUMT	1F	532.8	135.72	B
	SQP	1F	996.5	134.76	A
	OC	1F	3045.9	135.17	A
	FUD	-------	14.8	384.59	F

(d) For stress and displacement constraints

20. 252S,24D	SUMT	75S,1D	334.0	157.22	A
	SQP	75S,1D	628.8	156.95	A
	OC	17S	4342.2	225.96	D
	FUD	59S	184.5	179.38	C

(e) For displacement and frequency constraints

21. 24D,1F	SUMT	1D,1F	548.6	144.31	A
	SQP	1D,1F	1075.5	143.76	A
	OC	1D,1F	5290.8	143.93	A
	FUD	1D,1F	15.5	384.59	F

(f) For stress, displacement, and frequency constraints

22. 252S,3D,1F	SUMT	44S,1F	421.5	172.99	A
	SQP	46S,1F	739.9	172.68	A
	OC	3S	8489.5	269.49	F
	FUD	1F	633.3	225.57	D

TABLE 20.—DESIGN OF STIFFENED RING IDEALIZED BY MEMBRANE ELEMENTS ONLY
[24 design variables.]

(a) For stress constraints

Problem and constraints	Optimiza-tion method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
23. 72S	SUMT	28S	64.7	148.18	A
	SQP	28S	26.3	148.09	A
	OC	28S	121.6	148.18	A
	FUD	28S	9.9	148.22	A

(b) For displacement constraints

Problem and constraints	Optimiza-tion method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
24. 3D	SUMT	1D	62.0	136.32	B
	SQP	1D	102.9	133.53	A
	OC	1D	228.3	135.49	B
	FUD	1D	172.0	172.03	D

(c) For frequency constraints

Problem and constraints	Optimiza-tion method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
25. 1F	SUMT	1F	195.0	134.16	A
	SQP	1F	300.5	134.06	A
	OC	1F	843.9	134.02	A
	FUD	-------	10.3	302.45	F

(d) For stress and displacement constraints

Problem and constraints	Optimiza-tion method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
26. 72S,24D	SUMT	28S,1D	82.0	148.28	A
	SQP	27S,1D	33.8	148.18	A
	OC	18S,1D	1159.9	163.87	C
	FUD	28S,1D	11.4	148.46	A

(e) For displacement and frequency constraints

Problem and constraints	Optimiza-tion method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
27. 24D,1F	SUMT	1D,1F	255.0	143.30	A
	SQP	1D,1F	815.1	143.23	A
	OC	1D,1F	980.3	143.35	A
	FUD	-------	10.6	302.45	F

(f) For stress, displacement, and frequency constraints

Problem and constraints	Optimiza-tion method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
28. 72S,3D,1F	SUMT	17S,1F	202.2	168.46	A
	SQP	17S,1F	171.2	168.41	A
	OC	16S,1F	3758.5	177.68	B
	FUD	1F	118.7	218.41	D

TABLE 21.—DESIGN OF INTERMEDIATE–COMPLEXITY WING FOR STRESS, DISPLACEMENT, AND FREQUENCY CONSTRAINTS
[57 linked design variables.]

Problem and constraints	Optimization method	Number of active constraints	CPU time, sec	Optimum weight, lb	Grade
29. 316S,4D,1F	SUMT	28S	30 172.2	388.47	A
	SQP	-------	1 671.2	828.41	F
	OC	20S	30 879.9	988.70	F
	FUD	28S	9.9	454.19	C

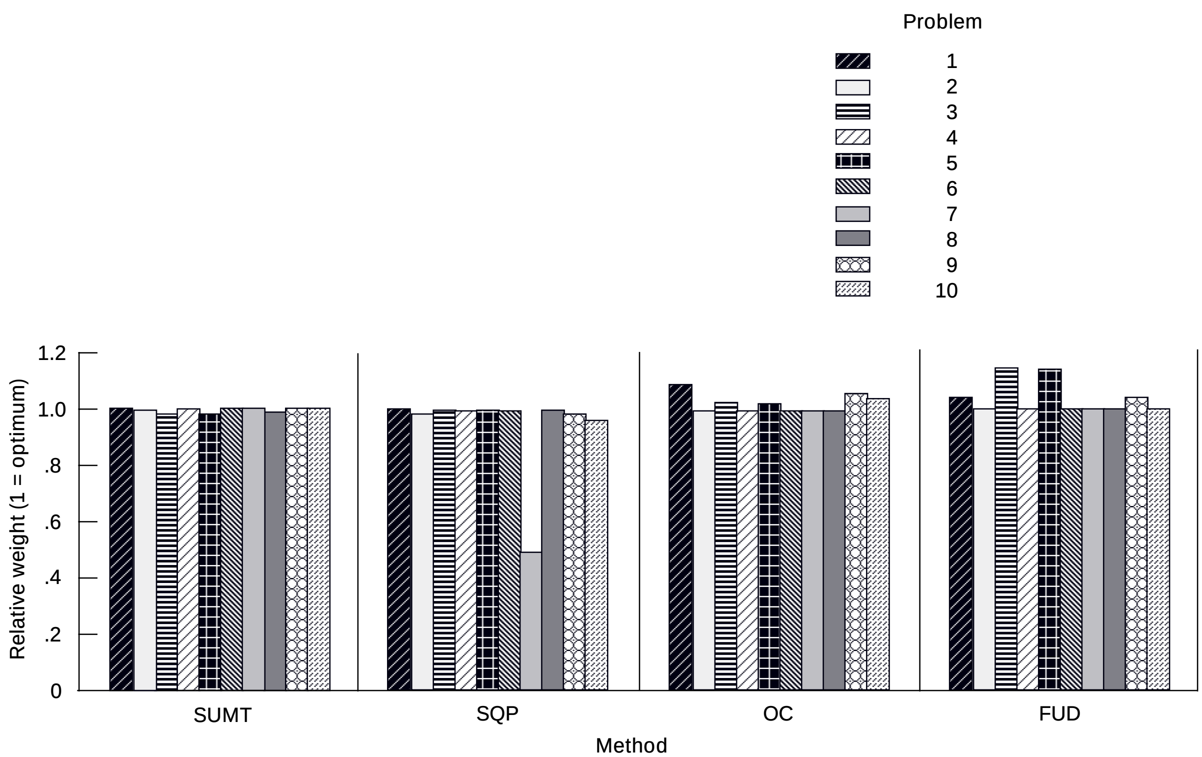


Figure 11.—Performance of optimization methods for design variable range 10 to 20. (Weight below 1.0 indicates infeasible design; weight above 1.0 represents overdesign. Method failed.)

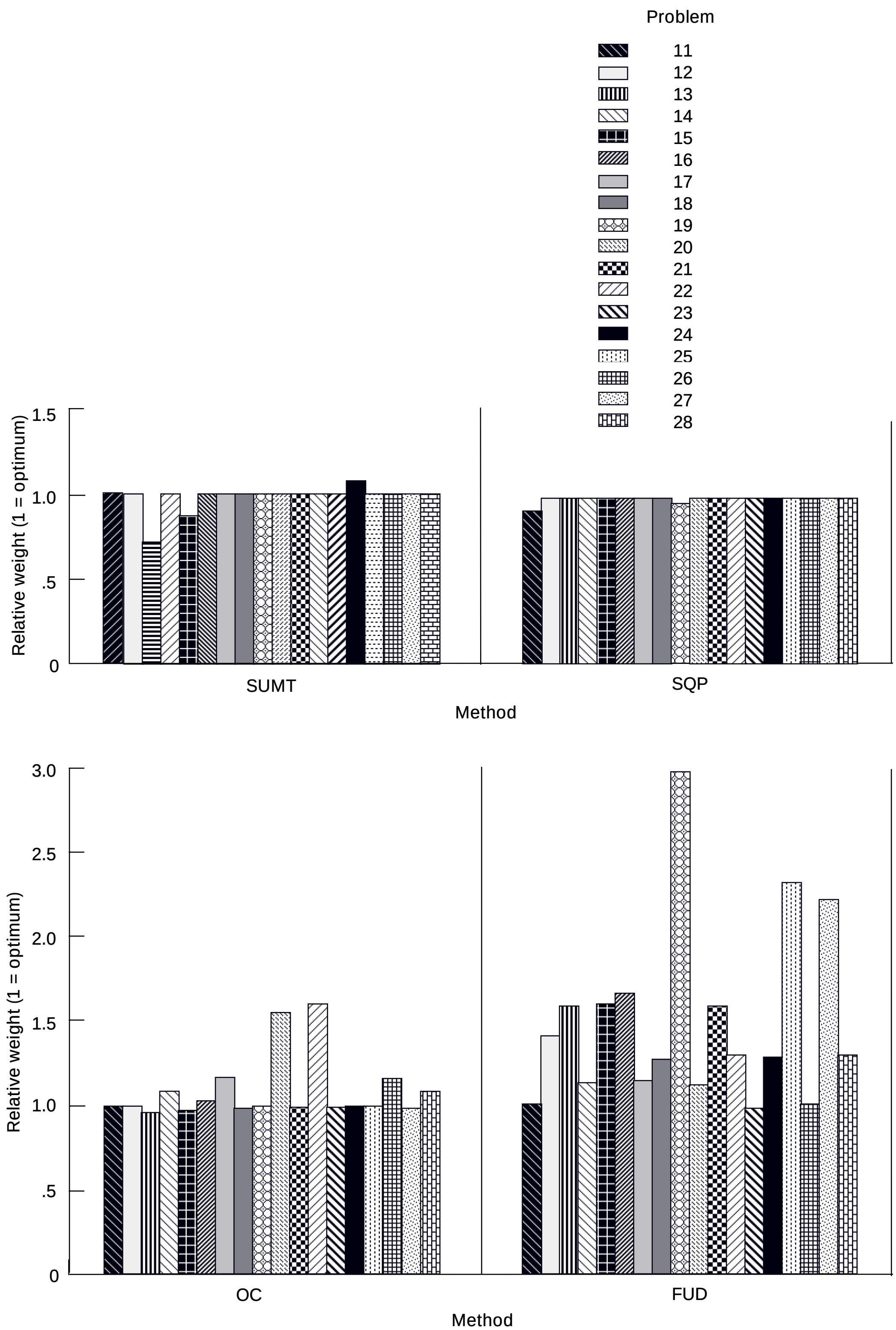


Figure 12—Performance of optimization methods for design variable range 21 to 50. (Weight below 1.0 indicates infeasible design; weight above 1.0 represents overdesign. Method failed.)

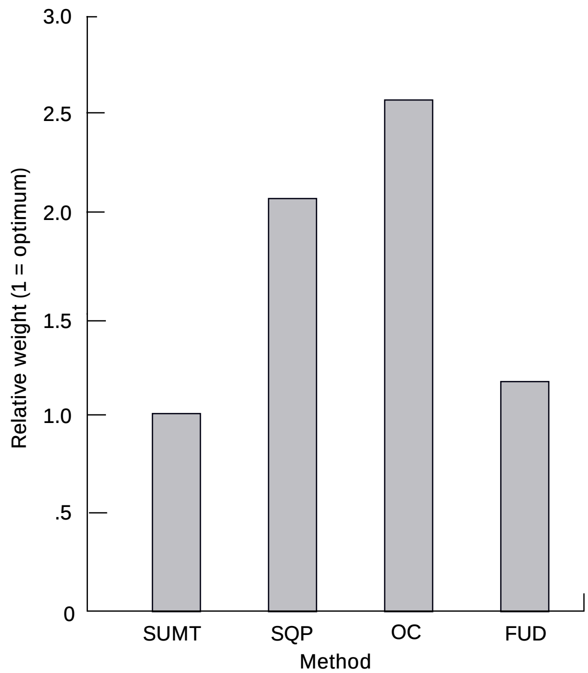


Figure 13.—Performance of optimization methods for 57 design variables (problem 29). (Weight below 1.0 indicates infeasible design; weight above 1.0 represents overdesign. Method failed.)

Appendix B

Glossary

CFA	closed-form analysis
DISP	ANALYZE/DANLYZE stiffness method
FD	modified method of feasible directions
FUD	fully utilized design concept
IFM	integrated force method
IMSL	quadratic programming algorithm from IMSL, Inc., Fortran subroutines
OC	optimality criteria methods
SIFMD	integrated force method where only displacement sensitivities are simplified
SIFMS	integrated force method where only stress sensitivities are simplified
SIFMSD	integrated force method where both stress and displacement sensitivities are simplified
SLP	sequential linear programming technique
SQP	sequential quadratic programming technique
SUMT	sequence of unconstrained minimizations technique

References

1. Miura, H.; and Schmit, L.A., Jr.: NEWSUMT—A Fortran Program for Inequality Constrained Function Minimization—Users Guide. Grant NGR–05–007–337, California University, Los Angeles, CA, June 1979. (Avail. CASI.)
2. DOT User’s Manual. Version 2.00. Engineering Design Optimization, Inc., Santa Barbara, CA, 1989.
3. Schittkowski, K.: NCONG/DNCONG: IMSL Math/Library User’s Manual FORTRAN Subroutines for Mathematical Applications. Version 2.0. International Mathematical Subroutines Library, Houston, TX, 1991.
4. Arora, J.S.: IDESIGN User’s Manual Version 3.5.2. Optimal Design Laboratory, The University of Iowa, Iowa City, IA, June 1989.
5. Patnaik, S.; Guptill, J.D.; and Berke, L.: Merits and Limitations of Optimality Criteria: Method for Structural Optimization. NASA TP–3373, 1993.
6. Gallagher, R.H.; and Zienkiewicz, O.C., eds.: Optimum Structural Design. John Wiley & Sons, 1973.
7. Venkayya, V.B.; and Tischler, V.A.: ANALYZE: Analysis of Aerospace Structures With Membrane Elements. Report AFFDL–TR–78–170, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, 1978.
8. Patnaik, S.N.; et al.: Improved Accuracy for Finite Element Structural Analysis via a New Integrated Force Method. NASA TP–3204, 1992.
9. Patnaik, S.N.; Hopkins, D.A.; and Coroneos, R.M.: Structural Optimization With Approximate Sensitivities. NASA TM–4553, 1994.
10. Haftka, R.T.; and Kamat, M.P.: Elements of Structural Optimization. Kluwer Academic Publishers, Boston, MA, 1985.
11. Schmit, L.A., Jr.: Structural Design by Systematic Synthesis. Proceedings of the 2nd Conference on Electronic Computation, ASCE, New York, 1960, pp. 105–132.
12. Zoutendijk, G.: Methods of Feasible Directions: A Study in Linear and Nonlinear Programming. Elsevier, New York, 1960.
13. Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design: With Applications. McGraw Hill, New York, 1984.
14. Berke, L.; and Khot, N.S.: Use of Optimality Criteria Methods for Large Scale Systems. Structural Optimization, AGARD Lecture Series 70, AGARD–LS–70, 1974.
15. Khot, N.S.; Venkayya, V.B.; and Berke, L.: Optimum Design of Composite Structures With Stress and Deflection Constraints. AIAA Paper 75–141, 1975.
16. The NASTRAN User’s Manual (Level 17.5). NASA SP–222(05), 1978.
17. Patnaik, S.N.; and Gallagher, R.H.: Gradients of Behavior Constraints and Reanalysis via the Integrated Force Method. Int. J. Numer. Methods Eng., vol. 23, 1986, pp. 2205–2215.
18. Garvey, S.J.: The Quadrilateral Shear Panel. Aircraft Eng., vol. 23, no. 267, May 1951, pp. 134–135.
19. Kiusalass, J.; and Reddy, G.B.: DESAP1, A Structural Design Program With Stress and Displacement Constraints, Vol. II: Sample Problems. NASA CR–2795, 1977.

Subject Index

E

F

G

S

T

U

V

W

X

Y

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)

2. REPORT DATE
October 1996

3. REPORT TYPE AND DATES COVERED
Technical Memorandum

4. TITLE AND SUBTITLE
CometBoards Users Manual
Release 1.0

5. FUNDING NUMBERS
WU–505–63–5B

6. AUTHOR(S)
James D. Guptill, Rula M. Coroneos, Surya N. Patnaik, Dale A. Hopkins, and Laszlo Berke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135–3191

8. PERFORMING ORGANIZATION REPORT NUMBER
E–8130

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, D.C. 20546–0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TM–4537

11. SUPPLEMENTARY NOTES
James D. Guptill, Rula M. Coroneos, Dale A. Hopkins, and Laszlo Berke, NASA Lewis Research Center; Surya N. Patnaik, Ohio Aerospace Institute, 22800 Cedar Point Road, Brook Park, Ohio 44142. Responsible person, Dale A. Hopkins, organization code 5210, (216) 433–3260.

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified - Unlimited
Subject Category 39

This publication is available from the NASA Center for AeroSpace Information, (301) 621–0390.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 words)***

Several nonlinear mathematical programming algorithms for structural design applications are available at present. These include the sequence of unconstrained minimizations technique, the method of feasible directions, and the sequential quadratic programming technique. The optimality criteria technique and the fully utilized design concept are two other structural design methods. A project was undertaken to bring all these design methods under a common computer environment so that a designer can select any one of these tools that may be suitable for his/her application. To facilitate selection of a design algorithm, to validate and check out the computer code, and to ascertain the relative merits of the design tools, modest finite element structural analysis programs based on the concept of stiffness and integrated force methods have been coupled to each design method. The code that contains both these design and analysis tools, by reading input information from analysis and design data files, can cast the design of a structure as a minimum-weight optimization problem. The code can then solve it with a user-specified optimization technique and a user-specified analysis method. This design code is called CometBoards, which is an acronym for “Comparative Evaluation Test Bed of Optimization and Analysis Routines for the Design of Structures.” This manual describes for the user a step-by-step procedure for setting up the input data files and executing CometBoards to solve a structural design problem. The manual includes the organization of CometBoards; instructions for preparing input data files; the procedure for submitting a problem; illustrative examples; and several demonstration problems. A set of 29 structural design problems have been solved by using all the optimization methods available in CometBoards. A summary of the optimum results obtained for these problems is appended to this users manual. CometBoards, at present, is available for Posix-based Cray and Convex computers, Iris and Sun workstations, and the VM/CMS system.

14. SUBJECT TERMS
Structural analysis; Nonlinear programming; Linear programming; Optimization; Test bed; Displacement method; Force method; Structural design criteria

15. NUMBER OF PAGES
166

16. PRICE CODE
A08

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	