

Structural Optimization With Approximate Sensitivities

S.N. Patnaik
Ohio Aerospace Institute
Brook Park, Ohio

D.A. Hopkins and R. Coroneos
Lewis Research Center
Cleveland, Ohio



National Aeronautics and
Space Administration

Office of Management

Scientific and Technical
Information Program

1994

Structural Optimization With Approximate Sensitivities

S.N. Patnaik
Ohio Aerospace Institute
Brook Park, Ohio

D.A. Hopkins
Lewis Research Center
Cleveland, Ohio

R. Coroneos
Lewis Research Center
Cleveland, Ohio

Summary

Computational efficiency in structural optimization can be enhanced if the intensive computations associated with the calculation of the sensitivities, that is, gradients of the behavior constraints, are reduced. Approximation to gradients of the behavior constraints that can be generated with small amount of numerical calculations is proposed. Structural optimization with these approximate sensitivities produced correct optimum solution. Approximate gradients performed well for different nonlinear programming methods, such as the sequence of unconstrained minimization technique, method of feasible directions, sequence of quadratic programming, and sequence of linear programming. Structural optimization with approximate gradients can reduce by one third the CPU time that would otherwise be required to solve the problem with explicit closed-form gradients. The proposed gradient approximation shows potential to reduce intensive computation that has been associated with traditional structural optimization.

Introduction

Structural optimization, via nonlinear mathematical programming techniques, follows two distinct steps. First, a search direction is generated and then a move distance along that direction is determined. The direction and the move distance are used to update and iterate the design until convergence to an optimum. The generation of a search direction typically requires the gradients of the behavior constraints of the structural optimization problem. Gradients of the behavior constraints can be obtained either in explicit terms by repeated application of the chain rule of differentiation or numerically by using a finite-difference scheme. In both cases computationally intensive gradient calculations are required. The primary goal of this study was to explore the possibility of improving computational efficiency of calculations of the gradients of the behavior constraints, thereby making optimization less computation intensive.

Gradient of behavior constraints, such as stresses and displacements, can be considered to consist of two distinct terms. The first accounts for the local effects, while the second represents the influence of the entire structure on the gradient. The first term is easy to calculate, while the second term is computation intensive. The proposition, here, is to retain the first term and to explore whether an approximate gradient will suffice for the second, that is, whether the optimum can be reached with fewer calculations. Even though different nonlinear programming techniques, such as methods of feasible direction, sequential quadratic programming, and penalty function, etc., use gradient information to calculate search directions, the actual implementations depend on the optimizer chosen. The feasibility of using approximate gradients in structural optimization will be investigated for different optimization methods to ensure that the conclusions are independent of the optimization algorithms. Optimization using approximate gradients has been incorporated into the design code CometBoards (which stands for Comparative Evaluation Test Bed of Optimization and Analysis Routines for the Design of Structures (ref. 1)). Many optimization problems have been solved successfully. For numerical illustration, two examples were considered: a forward swept wing and a ring structure. In addition, summaries of the results for several other problems are given without detailed elaboration.

The subject matter of this paper is presented in six sections: design as a nonlinear programming problem and solution methods, optimization methods, approximate sensitivities of behavior constraints, numerical illustrations, discussions, followed by conclusions.

Design As Nonlinear Programming Problem And Solution Methods

To examine the merits and limitations of approximate gradients in structural optimization, the design of trusses under multiple load conditions is cast as the following nonlinear

mathematical programming problem and solved using different algorithms:

Find the n design variables, such as member areas of a truss within prescribed upper and lower bounds that make the scalar weight function W a minimum under a set of stress and displacement constraints.

The weight function W can be written as

$$W = \sum_{j=1}^n w_j A_j L_j \quad (1)$$

where A_j is the cross sectional area; L_j , the length; and w_j , the weight density of the j^{th} element. To reduce the number of independent design variables, areas of a group of members are linked; thus the weight function defined by equation (1) has to be modified. However, this modification is carried out automatically and, because of its complexity, is not shown here in explicit terms.

To evaluate the effect of gradient approximations, only stress and displacement constraints are considered. These constraints can be formulated as

Stress constraint—

$$g_{\sigma_i} = \left| \frac{\sigma_i}{\sigma_{io}} \right| - 1 \leq 0 \quad (i = 1, \dots, n_s) \quad (2)$$

Displacement constraint—

$$g_{x_j} = \left| \frac{x_j}{x_{jo}} \right| - 1 \leq 0 \quad (j = n_s + 1, \dots, n_s + n_d) \quad (3)$$

where σ_i is the design stress for the i^{th} element, σ_{io} is the permissible stress for the i^{th} element; x_j is the j^{th} displacement component, x_{jo} is the displacement limitation for the j^{th} displacement component, and n_s and n_d are the number of stress and displacement constraints, respectively.

In a mathematical programming technique, the optimal design \bar{x}^{opt} is obtained iteratively from an initial design \bar{x}^0 in, say, K iterations. The design is updated at each iteration (k) by calculating two quantities: a direction \bar{S}_k and a step length α_k . The optimal design, using the direction and associated step length, can be written as

$$\bar{x}^{\text{opt}} = \bar{x}^0 + \sum_{k=1}^K \alpha_k \bar{S}_k \quad (4)$$

The direction \bar{S}_k is typically generated from the gradients of the objective function and the gradients of the active constraints. A one-dimensional search along the direction \bar{S}_k is carried out to obtain the optimum move distance α_k . The design is updated, and iteration is continued until convergence or until a stop criterion is satisfied.

Optimization Methods

In this investigation, four different optimization methods were considered: (1) sequence of unconstrained minimizations technique (SUMT), (2) sequential quadratic programming technique of the International Mathematical Subroutine Library (IMSL) routine DNCONG, (IMSL-SQP) (3) method of feasible directions (FD), and (4) sequence of linear programming (SLP). The selected optimization techniques are well known in the literature, and hence only a brief description of each method is provided herein. Readers, however, may refer to specific references for further details on each optimization method.

Sequence of Unconstrained Minimization Technique

In the sequence of unconstrained minimization technique (SUMT), the constrained optimization problem is solved as a sequence of unconstrained minimization problems through an extended penalty function (ref. 2). The direction vector \bar{S}_k in SUMT is calculated from the gradients of the behavior constraints and objective function following a modified Newton's approach.

Sequential Quadratic Programming Technique, DNCONG of IMSL

The sequential quadratic programming method, available in IMSL DNCONG routine (IMSL-SQP), solves the nonlinear problem as a sequence of quadratic subproblems (ref. 3). The direction vector in IMSL-SQP is generated by solving a quadratic subproblem with a quadratic approximation of the objective function and a linearization of the behavior constraints. The constraint linearization requires the gradients of the behavior constraints.

Method of Feasible Directions

In the method of feasible directions (FD) a search direction \bar{S} is determined that simultaneously satisfies two conditions: (1) the direction is feasible, that is, $\bar{S}^T \nabla_{g_j} < 0$, and (2) the direction is usable, that is, $\bar{S}^T \nabla W < 0$. Here, ∇_{g_j} represents the gradients of active constraints. Further information of the method of feasible directions can be found in references 4 to 6.

Sequential Linear Programming

In the method of sequential linear programming (SLP), an SLP for the original, nonlinear problem is obtained by linearizing a set of critical constraints and the objective function about a design point by Taylor series approximation (ref. 4). The linearization uses the gradients of the behavior constraints and objective function.

Approximate Sensitivities of Behavior Constraints

Sensitivity matrices of behavior constraints, such as stresses and displacements, can be considered to have two distinct terms: The first, which accounts for local effects, can be calculated with minimal computational effort once the structural analysis has been completed. The second term, which pertains to the response of the total structure as a single unit, is much more complex and requires extensive numerical computation. The calculations for both terms have been published earlier (refs. 7 and 17) and will not be repeated here. In this study the closed-form analytical sensitivity expressions were specialized to generate approximate gradients of stresses and displacements. The complexity of each term of the sensitivity expression, is illustrated for a three-bar truss.

Because an analysis tool has little influence on the performance of a nonlinear programming algorithm, either the force method or the displacement method can be used to illustrate the basic concepts. The integrated force method (IFM) of analysis was used to examine the merits and limitations of approximate gradients since it had been used earlier to formulate explicit sensitivities (ref. 7), and it brings simplicity and clarity to gradient expressions. The IFM (refs. 8 and 9) considers all internal forces $\{F\}$ as the primary unknowns, which are obtained as the solution to a system of governing equations. These governing equations are obtained by coupling the m equilibrium equations ($[B]\{F\} = \{P\}$) to the ($r = n - m$) compatibility conditions ($[C][G]\{F\} = \{0\}$) as

$$\begin{bmatrix} [B] \\ [C][G] \end{bmatrix} \{F\} = \begin{bmatrix} P \\ 0 \end{bmatrix}; [S]\{F\} = \{P^*\} \quad (5)$$

where $[B]$ is the ($m \times n$) equilibrium matrix, $[C]$ is the ($r \times n$) compatibility matrix, $[G]$ is the ($n \times n$) concatenated block diagonal flexibility matrix, $\{F\}$ is the n component internal force vector, $\{P\}$ is the m component external mechanical load vector, $[S]$ is the ($n \times n$) IFM governing matrix, n is the internal force degrees of freedom of the structure, and m is its displacement degrees of freedom. Displacements can be obtained from forces $\{F\}$ by back substitution as

$$\{X\} = [J][G]\{F\} \quad (6)$$

where $\{X\}$ is the m component nodal displacement vector and $[J] = m$ rows of $[S]^{-1}T$.

The IFM provides two basic equations, one for forces (eq. (5)) and the other for displacements (eq. (6)). The force equation can be differentiated to obtain the sensitivity of stress parameters, and the displacement equation can be differentiated for displacement sensitivity. The sensitivity expressions, which had been formulated previously (ref. 7), are not repeated herein, but their final forms are given.

Explicit Gradients of Stresses

The gradient of stresses $[\nabla\sigma]$ is a ($n \times n$) matrix. Its n rows correspond to n design variables, representing the areas of the n members of a truss, and its n columns represent the n member stress components.

$$[\nabla\sigma] = [\{\nabla\sigma_1\}, \{\nabla\sigma_2\}, \dots, \{\nabla\sigma_n\}] \quad (7)$$

where the gradient of a stress component $\{\nabla\sigma_k\}$ is

$$\{\nabla\sigma_k\} = \begin{bmatrix} \frac{\partial}{\partial A_1} \\ \frac{\partial}{\partial A_2} \\ \vdots \\ \frac{\partial}{\partial A_n} \end{bmatrix} \sigma_k \quad (8)$$

With these definitions, the explicit ($n \times n$) gradient matrix for stresses $\{\sigma\}$ can be written as (ref. 7)

$$[\nabla\sigma] = \left[\frac{-F}{A^2} \right] + [D]^T \left[\frac{1}{A} \right] \quad (9a)$$

$$[D] = [S]^{-1} \begin{bmatrix} [0] \\ [C][G] \end{bmatrix} [\bar{F}] \quad (9b)$$

where

$$[\bar{G}] = \begin{bmatrix} \bar{g}_{11} & & & \\ & \bar{g}_{22} & & \\ & & \ddots & \\ & & & \bar{g}_{nn} \end{bmatrix}$$

$$\left[\frac{1}{A} \right] = \begin{bmatrix} \frac{1}{A_1} & & & \\ & \frac{1}{A_2} & & \\ & & \ddots & \\ & & & \frac{1}{A_n} \end{bmatrix}$$

$$\bar{g}_{kk} = \frac{g_{kk}}{A_k}; \quad g_{kk} = \left(\frac{\ell}{AE} \right)_k$$

$$[\bar{F}] = \begin{bmatrix} F_1 & & & \\ & F_2 & & \\ & & \ddots & \\ & & & F_n \end{bmatrix}$$

$$\left[\frac{-F}{A^2} \right] = \begin{bmatrix} -F_1/A_1^2 & & & \\ & -F_2/A_2^2 & & \\ & & \ddots & \\ & & & -F_n/A_n^2 \end{bmatrix}$$

The first term in stress gradients (eq. (9a)) is a simple diagonal matrix representing local contributions, and its computation requires a trivial amount of calculations after analysis has been completed for forces $\{F\}$. The second term, however, is more complex, representing contributions to the sensitivity from the entire structure, and its calculation is computation intensive.

Explicit Gradients of Displacements

The gradients of displacements $[\nabla X]$ is a $(n \times m)$ matrix. Its columns correspond to m displacement components, and its rows represent the n design variables. The sensitivity matrix of the displacement constraints can be written as (ref. 7)

$$[\nabla X] = [\{\nabla X_1\}, \{\nabla X_2\}, \dots, \{\nabla X_n\}] \quad (10a)$$

$$[\nabla X] = [J][S_{dg}]^T + [J][G][D] \quad (10b)$$

The elements of the diagonal matrix $[S_{dg}]$ are given by

$$(S_{dg})_{kk} = -g_{kk} F_k / A_k \quad (11)$$

The first term in equation (10b) has a sparse form and requires trivial computation, while the second term can be computation intensive. Analytical gradient expressions given by equations (9a) and (10b) are hereinafter referred to as IFM explicit gradients.

Approximate Gradients of Stresses

The approximate gradient of stresses, which is a diagonal matrix of dimension $(n \times n)$, is the first term in equation (9a),

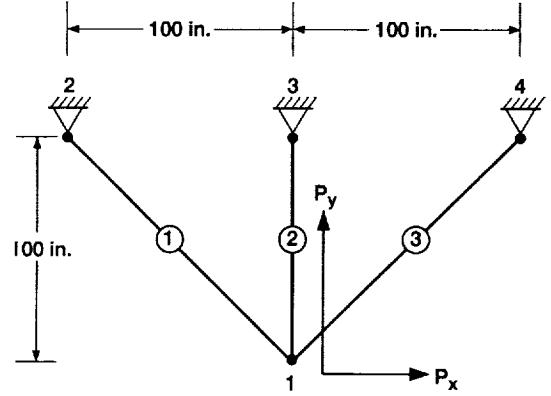


Figure 1.—Three-bar truss. (Elements are circled, nodes are not.)

and it has the following form:

$$[\tilde{\nabla} \sigma] = \left[\frac{-F}{A^2} \right] \quad (12)$$

where $[\tilde{\nabla} \sigma]$ is the approximate gradient of stresses. From equation (12), it can be observed that the calculation of approximate gradients of stresses requires minimal computations once analysis has been completed for forces.

Approximate Gradients of Displacements

Approximate gradients of displacements $[\tilde{\nabla} X]$, obtained by retaining the first term in equation (10b), has the following form:

$$[\tilde{\nabla} X] = [J][S_{dg}]^T \quad (13)$$

The matrix $[J]$ is calculated during the determination of displacements (see eq. (6)), and $[S_{dg}]$ is a diagonal matrix; thus, the calculation of displacement sensitivity also involves minimal computations. The approximate gradient expressions given by equations (12) and (13) are referred to as IFM approximate gradients.

Gradients for a Three-Bar Truss

To illustrate the complexity of each term in the gradient expression, an example of a three-bar truss shown in figure 1 was considered. The three member stresses ($\sigma_1, \sigma_2, \sigma_3$) and two nodal displacements (X_1, X_2) were considered to be the behavior constraints, and their sensitivities were calculated for the three member areas (A_1, A_2, A_3). The closed-form analytical gradient of member stresses has the following form:

$$[\nabla\sigma] = \begin{bmatrix} -F_1/A_1^2 & 0 & 0 \\ 0 & -F_2/A_2^2 & 0 \\ 0 & 0 & -F_3/A_3^2 \end{bmatrix} + \begin{bmatrix} \Re_1 e_{11} & \Re_1 e_{12} & \Re_1 e_{13} \\ \Re_2 e_{21} & \Re_2 e_{22} & \Re_2 e_{23} \\ \Re_3 e_{31} & \Re_3 e_{32} & \Re_3 e_{33} \end{bmatrix} \quad (14)$$

where

$$\Re_1 = \frac{\ell_1 F_1}{\sqrt{2} A_1^2 E}; \quad \Re_2 = \frac{-\ell_2 F_2}{A_2^2 E}; \quad \Re_3 = \frac{\ell_3 F_3}{\sqrt{2} A_3^2 E}$$

$$e_{11} = \frac{1}{r} [\sqrt{2} \rho_1 (\rho_2 + \rho_3)]; \quad e_{12} = \frac{1}{r} [-\sqrt{2} \rho_1 \rho_3];$$

$$e_{13} = \frac{1}{r} [-\sqrt{2} \rho_1 \rho_2 \rho_3]$$

$$e_{21} = \frac{1}{r} [\rho_2 (\rho_3 - \rho_1)]; \quad e_{22} = \frac{1}{r} [-\sqrt{2} \rho_1 + \rho_3];$$

$$e_{23} = \frac{1}{r} [-\sqrt{2} \rho_1 \rho_2 \rho_3]$$

$$e_{31} = \frac{1}{r} [-\sqrt{2} \rho_3 (\rho_1 + \rho_2)]; \quad e_{32} = \frac{1}{r} [-\sqrt{2} \rho_1 \rho_3];$$

$$e_{33} = \frac{1}{r} [\sqrt{2} \rho_1 \rho_2 \rho_3]$$

and where $r = \rho_1 \rho_2 + 2\rho_1 \rho_3 + \rho_2 \rho_3$, $\rho_i = A_i E_i / \ell_i$, and A_i , E_i , ℓ_i (for $i = 1, 2, 3$) are the areas, Young's moduli, and lengths of each of the three-bar elements of the truss, respectively.

The approximate gradients of the stress constraints has the form

$$\{\tilde{\nabla}\sigma_1\} = \begin{bmatrix} -\frac{F_1}{A_1^2} \\ 0 \\ 0 \end{bmatrix}, \quad \{\tilde{\nabla}\sigma_2\} = \begin{bmatrix} 0 \\ -\frac{F_2}{A_2^2} \\ 0 \end{bmatrix}, \quad \text{and} \quad \{\tilde{\nabla}\sigma_3\} = \begin{bmatrix} 0 \\ 0 \\ -\frac{F_3}{A_3^2} \end{bmatrix} \quad (15)$$

Note that approximate gradients of stresses given in equation (15) are much simpler than their closed-form gradients given in equation (14).

The approximate gradients for displacement constraints for the three-bar truss are

$$\{\tilde{\nabla}X_1\} = \left(\frac{\ell}{E}\right) \begin{bmatrix} 0 \\ -\frac{F_2}{A_2^2} \\ 0 \end{bmatrix}, \quad \text{and} \quad \{\tilde{\nabla}X_2\} = \left(\frac{\ell}{E}\right) \begin{bmatrix} -\frac{2F_1}{A_1^2} \\ -\frac{F_2}{A_2^2} \\ 0 \end{bmatrix} \quad (16)$$

Note that the displacement gradient expressions do not involve the third force component (F_3) because of the strain compatibility condition of the IFM (refs. 8 and 9). As before, the approximate displacement gradient expression given by equation (16) is much simpler than their closed-form given by equation (10b).

Numerical Illustrations

Optimum results for a set of 10 examples are provided. All examples were selected from structural optimization literature and are frequently used as test cases. For this illustration, only, two examples are presented from this set. The first quantifies the computations required to calculate constraints and their gradients. The second illustrates the merits and limitations of approximate gradients used in structural optimization. Results for the other examples in the set are summarized in tables 8 to 16 with brief discussions.

Numerical Example 1

The forward swept wing (ref. 10) depicted in figure 2 is used to quantify the computation time required to calculate constraint functions and their gradients. The wing was modeled by finite elements with 30 grid points and 135 truss elements. The structure is made of aluminum with a Young's modulus E of 10 000 ksi, a Poisson's ratio, ν of 0.3, and a weight density, w , of 0.1 lb/in.³. The structure has 135 design variables, being the areas of truss members, and 137 behavior constraints consisting of 135 stress and 2 displacement constraints. Further details of the problem that are not essential here can be found in references 10 and 11.

Constraint and gradient calculations for the problem were carried out in an SGI 4D/35 Unix workstation. Even though, theoretically, the CPU time required to calculate constraints and their gradients should remain constant, in actuality some variation in CPU time did occur because of factors such as system usage and memory utilization of the workstation. An average CPU time for each optimization was calculated by solving the problem several times. The computation time

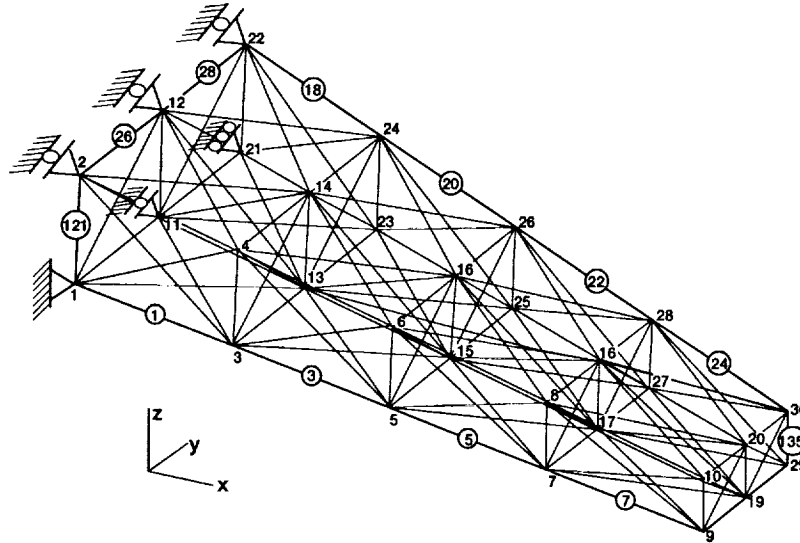


Figure 2.—Forward-swept wing. (Representative elements are circled, nodes are not.)

is normalized with respect to the CPU time required to calculate constraints when the problem is solved using the optimizer SUMT. The results, presented in table 1, show that the average CPU time to calculate constraints and gradients varies by less than 1 percent for the different optimization methods. On average, the CPU time required to calculate explicit gradients using the chain rule of differentiation is about 5 times more than that needed to calculate constraints. If numerical differentiation using a forward finite-difference scheme is followed, the gradient calculation can be about 22 times more than the calculation of the constraint functions. The time required to calculate approximate gradients takes about the same time that is required to calculate constraints. The actual difference in the CPU time to calculate constraints and their approximate gradients is very small.

Numerical Example 2

Minimum weight design of a 60-bar trussed ring under three multiple-load conditions for stress and displacement

constraints is considered to illustrate the use of approximate gradients in structural optimization (refs. 11 and 12). The problem is solved twice, first using explicit gradients and then using approximate gradients. To ensure accurate comparison, all conditions are kept identical except for the gradient expressions.

The ring shown in figure 3 has inner and outer radii of $R_i = 90$ in. and $R_o = 100$ in. The ring is idealized by 60 truss elements, and it is made of aluminum with Young's modulus E of 10 000 ksi and weight density w of 0.1 lb/in.³. The ring is subjected to three static-load conditions, as given in table 2. The constraints specified on stresses and displacements are given in table 3. The optimum design of the ring was determined using 198 behavior constraints, consisting of 180 stresses and 18 displacement constraints. The 60-bar cross sectional areas are linked to obtain a reduced set of 25 design variables (table 4). Minimum weight design for the ring was obtained first using closed-form explicit gradients. The design was obtained next using approximate gradients. In both cases the four optimization techniques described in the section "Optimization Methods" were used.

TABLE 1.—FORWARD SWEPT WING: COMPUTATION FOR CONSTRAINTS AND GRADIENTS

Optimization methods	Average CPU time		
	Constraints	Analytical gradients	Finite-difference gradients
SQP	0.983	4.967	21.163
FD	0.972	4.931	21.661
SLP	0.967	4.946	22.028
SUMT	^a 1.000	5.126	21.639

^aNormalized to unit for SUMT.

TABLE 2.—60-BAR TRUSSED RING: LOAD SPECIFICATIONS

Load condition	Node number	Load component, P_x , kip	Load component, P_y , kip
Case I	1	-10.0	0
	7	9.0	0
Case II	15	-8.0	3.0
	18	-8.0	3.0
Case III	22	-20.0	10.0

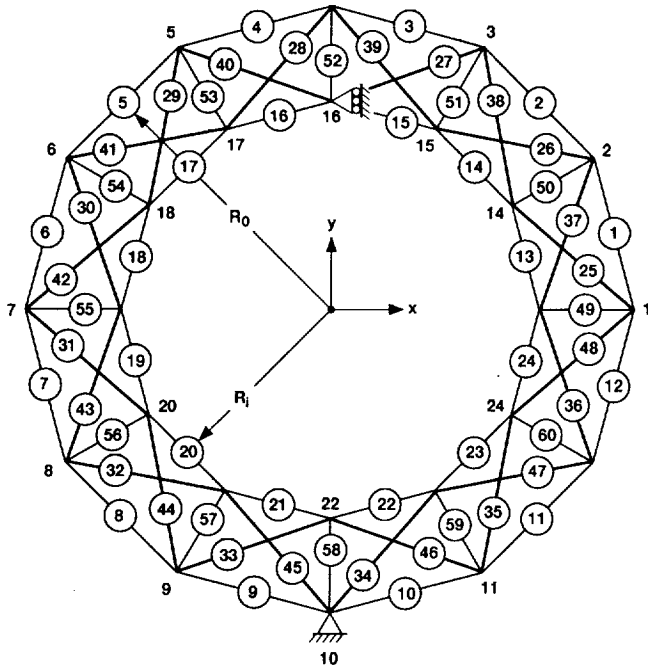


Figure 3.—Sixty-bar trussed ring. (Elements are circled, nodes are not.)

TABLE 3.—60-BAR TRUSSED RING:
CONSTRAINT SPECIFICATIONS

Constraint type	Constraint description
Stress	$\sigma_i \leq \sigma_o, i = 1, 2, \dots, 60$ $\sigma_o = 10 \text{ ksi}$
Displacement	Three constraints along both the x and y directions of magnitude: 1.75 in. at node 4, 2.25 in. at node 13, and 2.75 in. at node 19.
Minimum area	$A_i \geq 0.5 \text{ in.}^2$

TABLE 4.—60-BAR TRUSSED RING:
DESIGN VARIABLE LINKAGE

Serial number	Design variable	Member linked	Serial number	Design variable	Member linked
1	1	49 → 60	2	2	1, 13
3	3	2, 14	4	4	3, 15
5	5	4, 16	6	6	5, 17
7	7	6, 18	8	8	7, 19
9	9	8, 20	10	10	9, 21
11	11	10, 22	12	12	11, 23
13	13	12, 24	14	14	25, 37
15	15	26, 38	16	16	27, 39
17	17	28, 40	18	18	29, 41
19	19	30, 42	20	20	31, 43
21	21	32, 44	22	22	33, 45
23	23	34, 46	24	24	35, 47
25	25	36, 48			

The optimum designs and associated information obtained using explicit and approximate gradients are summarized in table 5. The rates of convergence of the weight versus the number of reanalyses for two optimization techniques (SUMT and IMSL-SQP) are presented in figures 4 and 5. From the information given in table 5 and figures 4 and 5, we observe the following:

Optimum weight.—The optimum weight of the ring is about 308 lb. This optimum is reached by all optimization methods when approximate or closed-form gradients are used with one exception: The SLP method converged to the correct optimum of 308.4 lb when approximate gradients were used, but yielded an approximately 1 percent over design (namely, 312 lb) when explicit closed-form gradients were used. Overall, the optimization with approximate gradients performed well for all four optimization methods.

Number of active constraints.—At optimum all optimization methods yielded one active displacement constraint when explicit or closed-form gradients were used. Optimization

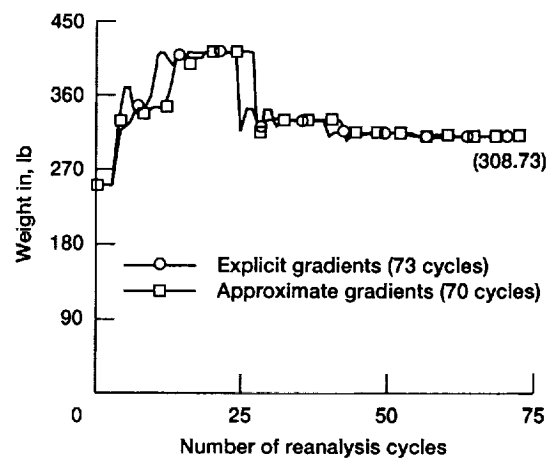


Figure 4.—Convergence of weight for ring using SUMT.

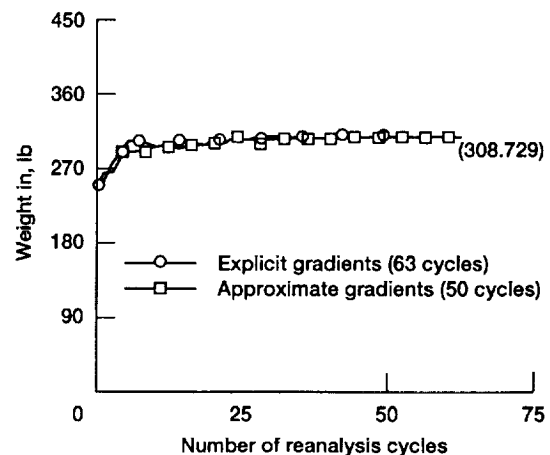


Figure 5.—Convergence of weight for ring using IMSL-SQP.

TABLE 5.—60-BAR TRUSSED RING: AVERAGE CPU TIME USING APPROXIMATE GRADIENTS

Optimization methods	Explicit gradients			Approximate gradients		
	Optimum weight, lb	Number of active constraints (a)	CPU time, ^b s	Optimum weight, lb	Number of active constraints (a)	CPU time, s
SUMT	308.730	24S, 1D	219.96	308.896	25S, 1D	51.41
IMSL-SQP	308.587	24S, 1D	170.58	308.729	25S, 1D	31.95
FD	308.406	24S, 1D	220.26	308.789	25S, 1D	97.91
SLP	312.024	23S, 1D	90.20	308.454	23S, 1D	51.27

^aS denotes active stress constraints; D denotes active displacement constraint.

^bCPU time on Convex minicomputer.

with approximate gradients yielded 25 stress constraints; with explicit gradients 24 stress constraints resulted. Optimization using approximate gradients performed well when there were numerous active constraints.

Amount of CPU time on Convex.—From table 5, it can be seen that the amount of computation time required when approximate gradients were used was considerably less than the CPU time required when explicit closed-form gradients were used. From a set of 29 examples that have been solved using different optimization methods (ref. 1), it was observed that IMSL-SQP is very reliable optimizer. For the ring problem, the IMSL-SQP with approximate gradients required only 18.7 percent of time required when explicit closed-form gradients were used. The percent of CPU time required to generate an optimum with gradient approximation was 23 percent by SUMT, 44 percent by FD, and 57 percent by SLP. The average CPU time required by all four methods when approximate gradients are used was about 33 percent of the CPU time required for the solution of the problem with closed-form analytical gradients. This reduction ratio is defined as

$$\eta = \frac{\text{CPU time with approximate gradients}}{\text{CPU time with closed form analytical gradients}}$$

Optimization with approximate gradients will be efficient if η is less than unity ($\eta < 1$). For the ring problem $\eta = 1/3$.

Number of reanalysis cycles.—Figures 4 and 5 show that the number of reanalysis cycles required to generate optimum solutions when approximate or explicit gradients were used was about the same but that somewhat fewer reanalysis cycles were required when approximate sensitivities were used. For example, for the ring problem, IMSL-SQP required 63 reanalysis cycles with closed-form sensitivities and only 50 with approximate sensitivities (see fig. 5). These values for the SUMT optimizer are 73 and 70, respectively

(see fig. 4). We had expected the number of function evaluations for the solution of a design problem to be higher with approximate sensitivities; however, this was not the case. Even though the number of reanalysis cycles to solve a problem with approximate and explicit gradients are comparable, the execution CPU times required for the two procedures are different because approximate calculations are inexpensive (see table 5).

Summary of Other Numerical Examples

Brief summaries of the nine examples, along with optimum results, are given in tables 6 to 16. The examples are presented under two groups. Structures with 25 or more members are in group I (table 6). Those with fewer than 25 members are listed in group II (table 7). The problem solution uses a linking strategy that reduces the number of independent design variables. (Hereinafter referred to as "linking," which is available in CometBoards (ref. 1).)

TABLE 6.—GROUP I: DESIGN OF LARGE STRUCTURES

Problem	Description	Constraint types	Number of design variables	Number of load conditions	Total number of constraints
1	135-Bar forward swept wing	Stress + Displacement	135	1	137
2	60-Bar trussed ring (linked)	Stress	25	3	180
3	41-Bar spacer truss (linked)	Stress + Displacement	1	1	56
4	25-Bar space truss (linked)	Stress + Displacement	8	2	86
5	10-Bay truss	Stress + Displacement	51	1	91

TABLE 7.— GROUP II: DESIGN OF SMALL STRUCTURES

Problem	Description	Constraint types	Number of design variables	Number of load conditions	Total number of constraints
6	10-Bar truss	Stress + Displacement	10	1	14
7	10-Bar truss (linked)	Stress + Displacement	5	1	14
8	5-Bar truss	Stress + Displacement	5	2	14
9	3-Bar truss	Stress + Displacement	3	1	5

Group I

- Problem 1 135-Bar forward swept wing (refs. 10 and 11).
Problem 2 60-Bar trussed ring with linking (refs. 11 and 12), (The constraint space for this problem is different from the ring discussed earlier in the section "Numerical Example 2.")
Problem 3 41-Bar spacer truss of the Space Station Freedom, with Linking (ref. 13).
Problem 4 25-Bar space truss with linking (ref. 14).
Problem 5 10-Bay truss (ref. 15).

Group II

- Problem 6 10-Bar truss (ref. 11).
Problem 7 10-Bar truss with linking (ref. 11).
Problem 8 5-Bar truss (refs. 11 and 16).
Problem 9 3-Bar truss (refs. 11 and 15).

These examples were solved using two nonlinear programming algorithms and three analysis tools: The algorithms were (1) IMSL-SQP, the constrained optimization routine, DNCONG, of IMSL, since this optimizer out performed most other methods (ref. 1); and (2) NPSOL, nonlinear programming solver routine E04UCF as implemented in the NAG Fortran library, which appears to be another efficient optimizer. The analysis tools were (1) the approximate IFM, (2) IFM, and (3) the displacement method (DISP) (ref. 17). The solutions were performed on a Cray YMP8/8128, version Unicos 7.0.4.4 computer. The CPU estimates on the Cray YMP are more approximate than on a personal SGI workstation because, at any given time, there may be several users and some operations maybe automatically carried out in parallel.

Optimum weight.—Tables 8 to 16 show that the approximate IFM provided correct optimum solutions for all problems when IMSL-SQP optimization algorithm was used. When NPSOL optimizer was used, only 8 of the 9 optimum solutions were correct. Problem 1 in group I yielded an overdesign of 8.6 percent when NPSOL is used. Also, NPSOL provided an infeasible design for problem 5 when the displacement method is used and an error of 19.7 percent in the optimum weight. However, the nonperformance for problem 1 when NPSOL is used does not appear to be a deficiency of the approximate sensitivity.

Amount of CPU Time on Cray-YMP.—The CPU time on the Cray-YMP can be approximate, especially for small problems. For large problems in group I, the average CPU time required to solve any one of the five problems using both IMSL-SQP and NPSOL optimization methods are 28.41 CPU for approximate IFM, 78.90 CPU for IFM, and 148.28 CPU for the displacement method. Overall, the nine examples performed as well as the ring problem, discussed in "Numerical Example 2". The CPU reduction ratio η for the ring was 33 percent on a Convex computer. The same reduction ratio η for the five problems in group I is 36

percent on a Cray-YMP. Note that when the displacement method is used as the analysis tool and sensitivities are calculated analytically in closed form, the reduction ratios are

$$\gamma_a = \frac{\text{CPU time by displacement analyzer}}{\text{CPU time by approximate IFM}} = 5.2$$

$$\gamma_b = \frac{\text{CPU time by displacement analyzer}}{\text{CPU time by IFM}} = 1.9$$

The displacement analyzer required twice the CPU time than needed by IFM and five times that needed by the approximate IFM. In other words the displacement analyzer, when compared with IFM or approximate IFM, appears to be an insufficient tool for structural optimization.

Discussions

Why do approximate gradients perform as well as the explicit closed-form gradients in structural optimization?

Although we do not know an exact answer to the question, we provide the following explanation: Consider the design of a structure with many design variables and many stress and displacement constraints as shown in figure 6. Select a small local segment of the structure, indicated by P in figure 6, and consider the internal forces at the local region at P . It can be assumed that the force variations at P depend on the relative values of the design variables within the local region surrounding P , along with such relative variations and contributions from other parts of the structure. As far as internal force variation is concerned, the local effects appear to dominate other influences in the structure. This observation has similar connotation with structural indeterminacy, that is, the effect of indeterminacy is neglected in the calculation of gradients. However, the effect of indeterminacy is used during the calculation of internal forces, which is represented through the bottom $(n-m)$ compatibility rows in the matrix $[S]$ (see eq. (5)). In brief, for the variation of stresses or the calculation of their approximate sensitivities the consideration of the local effects only may be sufficient in an iterative design optimization scheme.

Sensitivities of displacements do not follow the logic that is applicable to stresses because displacements are global variables. In determining displacement sensitivity, one must consider the effect of the total structure as a single unit. For example, the variation of displacement at point P cannot be considered to depend primarily on a local region surrounding the point P . In other words, displacements at point P can be easily influenced by changes in the flexibilities at a far-away location such as at a region Q which is close to the boundary. In approximate gradient calculation, the global nature of the displacement variable is represented through the

TABLE 8.—PROBLEM 1: FORWARD SWEEP WING FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb (a)	Normalized CPU time, ^b s	Number of gradient Calls	Normalized optimum weight, lb	Normalized CPU time, ^b s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	100	^c 1.086	0.26	100
IFM, explicit gradients	1.000	2.61	73	1.000	2.82	73
Displacement, explicit gradients	1.000	2.07	100	1.000	8.98	100

^aWeight is normalized with respect to IMSL-SQP results obtained using IFM approximate gradients.

^bCray YMP CPU time is normalized with respect to IMSL-SQP results obtained using IFM approximate gradients.

^cInfeasible design.

TABLE 9.—PROBLEM 2: 60-BAR TRUSSED RING WITH LINKING FOR STRESS CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	98	0.999	0.34	27
IFM, explicit gradients	1.000	0.31	14	1.000	2.26	74
Displacement, explicit gradients	1.000	0.48	13	1.000	1.72	49

^aCPU time on Cray YMP.

TABLE 10.—PROBLEM 3: 41-BAR SPACER TRUSS OF SPACE STATION FREEDOM FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	13	1.000	2.44	18
IFM, explicit gradients	1.000	2.49	13	1.000	4.27	18
Displacement, explicit gradients	1.000	4.51	12	1.000	3.44	28

^aCPU time on Cray YMP.

TABLE 11.—PROBLEM 4: 25-BAR SPACE TRUSS WITH LINKING FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	19	1.000	5.47	42
IFM, explicit gradients	1.000	1.69	19	1.000	7.25	48
Displacement, explicit gradients	1.000	2.41	19	1.000	26.00	393

^aCPU time on Cray YMP.

TABLE 12.—PROBLEM 5: 10-BAY TRUSS FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	73	0.994	3.83	453
IFM, explicit gradients	0.993	0.65	35	0.992	2.36	91
Displacement, explicit gradients	0.992	1.81	69	^b 0.797	0.96	20

^aCPU time on Cray YMP.

^bInfeasible design.

TABLE 13.—PROBLEM 6: 10-BAR TRUSS FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	17	1.013	6.52	48
IFM, explicit gradients	0.994	1.43	22	0.994	9.52	73
Displacement, explicit gradients	0.997	2.38	21	0.994	37.00	260

^aCPU time on Cray YMP.

TABLE 14.—PROBLEM 7: 10-BAR TRUSS WITH LINKING FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	18	1.000	6.25	38
IFM, explicit gradients	1.000	1.33	18	1.000	7.00	34
Displacement, explicit gradients	1.000	2.50	18	1.000	9.08	35

^aCPU time on Cray YMP.

TABLE 15.—PROBLEM 8: 5-BAR TRUSS FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	48	1.001	1.14	21
IFM, explicit gradients	1.000	0.52	21	1.000	1.59	19
Displacement, explicit gradients	1.000	0.90	17	1.000	2.00	26

^aCPU time on Cray YMP.

TABLE 16.—PROBLEM 9: 3-BAR TRUSS FOR STRESS AND DISPLACEMENT CONSTRAINTS

Analysis methods	IMSL-SQP			NAG-NPSOL		
	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls	Normalized optimum weight, lb	Normalized CPU time, ^a s	Number of gradient calls
IFM, approximate gradients	1.000	1.00	6	1.000	12.00	23
IFM, explicit gradients	1.000	1.00	7	1.000	7.00	10
Displacement, explicit gradients	0.989	3.00	6	1.000	12.50	20

^aCPU time on Cray YMP.

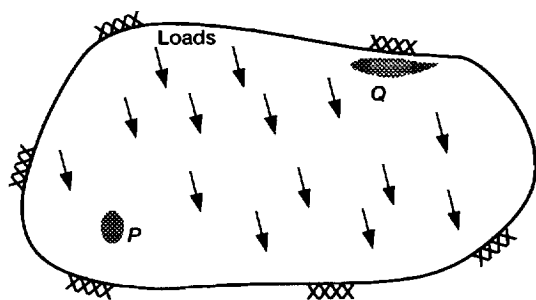


Figure 6.—A structural form.

influence coefficient matrix $[J] = m$ rows of $[[S]^{-1}]^T$. (See eqs. (6) and (13).) The matrix $[S]$ represents the effect of the entire structure as a single unit. Consider next a term of the diagonal matrix $(S_{dg})_{kk}$, which is used to calculate approximate sensitivities of displacements. This term can be factored as $(S_{dg})_{kk} = -g_{kk}(F_k/A_k) = -g_{kk}\sigma_k$, $\sigma_k = F_k/A_k$. As mentioned earlier, the stress terms can be approximated. In the approximate displacement sensitivity calculation, the global effect is retained through the displacement coefficient matrix $[J]$, and the effect of force variables is approximated. Such approximate gradients appear to be adequate in the design optimization with displacement constraints.

The approximation of gradients and the associated reduction in the amount of computations presented in this paper can be considered as an attempt to reduce numerical complexity in the design optimization of structural systems using nonlinear mathematical programming techniques. We believe gradient approximation is a fertile research avenue and that it should be extended to other types of common behavior constraints, such as frequency and stability constraints and other type of nontruss structures.

Conclusions

Calculation of explicit gradients or sensitivities of stress and displacement constraints required in structural optimization can be computationally intensive. Approximate sensitivities of such behavior constraints can be generated with trivial computational effort. Approximate sensitivities have been used to obtain correct optimum solutions to structural problems using different nonlinear optimization techniques, such as sequence of unconstrained minimizations technique, sequential quadratic programming, method of feasible directions, and sequential linear programming. Structural optimization with approximate gradients can reduce the CPU time required to solve a problem by about one-third the computation time required with explicit closed-form gradients.

Extension of the present research to frequency and stability constraints and to nontruss type structures can be fruitful.

Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio, January 14, 1994.

References

1. Guptill, J.D.; Coroneos, R.M.; Patnaik, S.N.; and Hopkins, D.: Comet-Boards User's Manual. Version 1.0, NASA Lewis Research Center, Cleveland, OH, 1993.
2. Miura, H.; and Schmit, Jr., L.A.: NEWSUMT—A Fortran Program for Inequality Constrained Function Minimization—User's Guide. NASA CR-159070, 1979.
3. Schittkowski, K.: NCONG/DNCONG: IMSL Math/Library User's Manual Fortran subroutines for Mathematical Applications. Version 2.0. International Mathematical Subroutine Library, Houston, TX, 1991.
4. DOT User's Manual, Version 2.00. Engineering Design Optimization, Inc., Santa Barbara, CA, 1989.
5. Zoutendijk, G.: Methods of Feasible Directions. Elsevier, New York, 1960.
6. Vanderplaats, G.N.: Numerical Optimization Techniques for Engineering Design: With Applications. McGraw Hill, New York, 1984.
7. 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.
8. Patnaik, S.N.; Hopkins, D.A.; Aiello, R.A.; and Berke, L.: Improved Accuracy for Finite Element Structural Analysis via a New Integrated Force Method. NASA TP-3204, 1992.
9. Patnaik, S.N.; Berke, L.; and Gallagher, R.H.: Integrated Force Method Versus Displacement Method for Finite Element Analysis. NASA TP-2937, 1990.
10. Venkayya, V.B.; Knot, N.S.; and Berke, L.: Application of Optimality Criteria Approach to Automated Structural Design of Large Practical Structures. Second Symposium on Structural Optimization, AGARD-CP-123, 1973, pp. 3-1 to 3-19.
11. Berke, L.; Guptill, J.D.; and Patnaik, S.N.: Merits and Limitations of Optimality Criteria Method for Structural Optimization. NASA TP-3373, 1993.
12. Berke, L.; Patnaik, S.N.; and Murthy, P.L.N.: Application of Artificial Neural Networks to the Design Optimization of Aerospace Structural Components. NASA TM-4389, 1993.
13. Gendy, A.; Patnaik, S.N.; Hopkins, D.; and Berke, L.: Preliminary Analysis and Design Optimization of the Short Spacer Truss of Space Station Freedom. NASA TM-4470, 1993.
14. Patnaik, S.N.; and Srivastava, N.K.: An Automated Optimum Design of Trusses. Computer Methods Appl. Mech. Eng., vol. 5, 1976, pp. 245–265.
15. Patnaik, S.N.; Guptill, J.D.; and Berke, L.: Singularities in Optimal Structural Design. NASA TM-4365, 1992.
16. Berke, L.; and Knot, N.S.: Use of Optimality Criteria Approach to Structural Optimization for Large Scale Systems. Structural Optimization, AGARD-LS-70, 1974, pp. 1–29.
17. Venkayya, V.B.; and Tischler, V.A.: ANALYZE—Analysis of Aerospace Structures with Membrane Elements. Report AFFDL-TR-78-170, Air Force Flight Dynamics Lab, Wright-Patterson, AFB, OH, 1978.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE January 1994	3. REPORT TYPE AND DATES COVERED Technical Memorandum	
4. TITLE AND SUBTITLE Structural Optimization With Approximate Sensitivities			5. FUNDING NUMBERS WU-505-63-5B	
6. AUTHOR(S) S.N. Patnaik, D.A. Hopkins, and R. Coroneos				
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-8226	
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-4553	
11. SUPPLEMENTARY NOTES S.N. Patnaik, Ohio Aerospace Institute, 22800 Cedar Point Road, Brook Park, Ohio 44142; and D.A. Hopkins and R. Coroneos, NASA Lewis Research Center. Responsible person, D.A. Hopkins, organization code 5210, (216) 433-3260.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Unclassified - Unlimited Subject Category 39			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 words) Computational efficiency in structural optimization can be enhanced if the intensive computations associated with the calculation of the sensitivities, that is, gradients of the behavior constraints, are reduced. Approximation to gradients of the behavior constraints that can be generated with small amount of numerical calculations is proposed. Structural optimization with these approximate sensitivities produced correct optimum solution. Approximate gradients performed well for different nonlinear programming methods, such as the sequence of unconstrained minimization technique, method of feasible directions, sequence of quadratic programming, and sequence of linear programming. Structural optimization with approximate gradients can reduce by one third the CPU time that would otherwise be required to solve the problem with explicit closed-form gradients. The proposed gradient approximation shows potential to reduce intensive computation that has been associated with traditional structural optimization.				
14. SUBJECT TERMS			15. NUMBER OF PAGES 16	
			16. PRICE CODE A03	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT	