AIAA JOURNAL
Vol. 30, No. 7, July 1992

# Sparse Approach to Simultaneous Analysis and Design of Geometrically Nonlinear Structures

Carlos E. Orozco* and Omar N. Ghattas†
*Carnegie Mellon University, Pittsburgh, Pennsylvania 15213*

**The problem of increasing the efficiency of the optimization process for nonlinear structures has been examined by several authors in the last ten years. One method proposed to improve the efficiency of this process considers the equilibrium equations as equality constraints of the nonlinear mathematical programming problem. The efficiency of this method, commonly called *simultaneous*, as compared to the more traditional approach of nesting the analysis and design phases, is reexamined in this paper. It is shown that, when projected Lagrangian methods are used, the simultaneous method is computationally more efficient than the nested, *provided* the sparsity of at least the Jacobian matrix is exploited. The basic structure of the Hessian and Jacobian matrices for geometrically nonlinear behavior of truss structures is given and numerical results are presented for a series of large problems using both dense and sparse projected Lagrangian methods.**

## I. Introduction

FINDING the optimum design of a structure, even when its topology and shape are predetermined, may be exceptionally expensive, especially for structures with a large number of degrees of freedom and/or elements. The problem is usually posed in such a way that the objective is to minimize the total weight or volume of the structure. The variables are element sizes that may be assigned individually or by groups. The constraints are usually established in terms of stresses and displacements, and are, therefore, nonlinear in nature. The resulting problem is then one of nonlinear mathematical programming, the solution of which requires the use of iterative techniques.

Conventionally, a structural analysis is first performed for some initial guess of the element sizes and its results are used to update the values of the objective function, the constraints, and their corresponding derivatives. Using these values, a search direction is determined with the aid of a nonlinear optimization algorithm. The search direction is then used to improve the current design, and the cycle is repeated until some convergence criterion is met. This approach is usually called *nested* and several procedures have been proposed to improve its performance.[1-4] The nonlinear case is particularly amenable to such an improvement because it is itself an iterative process. Wu and Arora,[3] for instance, use the sensitivity information of the behavior of the structure to provide a better initial guess for the next nonlinear analysis. Haftka[1] has extended this idea to include a linearized behavior of the structure at early optimization stages.

An alternative view of the overall design process, however, consists of considering the equilibrium equations of the system as equality constraints of the mathematical programming problem, and adding the displacements of the structure to the set of optimization variables. The resulting approach is usually termed *simultaneous*. Within the structural context this alternative was first proposed by Schmit and Fox[5,6] and has been used extensively by many authors in several different disciplines.[7-13] Recently, several attractive features of the approach have been discovered. Smaoui and Schmit[14] found for instance, that the simultaneous approach can effectively detect elastic instabilities for geometrically nonlinear structures when used in conjunction with a generalized reduced gradient algorithm. Also, Bendsoe et al.[15] and Ghattas and Schrader[16] have shown that the method can be used to advantage in topological structural optimization. In the present work we concentrate on the numerical aspects of the approach, specifically on its sparsity characteristics.

There are numerous techniques to solve nonlinear mathematical programming problems. It is now generally accepted, however, that projected Lagrangian methods constitute the most efficient tool to solve this class of problems.[17,18] Earlier techniques required the feasibility of the current approximation of the solution to be enforced at all times. When projected Lagrangian methods are used, feasibility of the estimate of the solution is only satisfied at the end of the optimization process. In a nested formulation, equilibrium is automatically satisfied at each optimization step because a complete nonlinear structural analysis is performed each time. If a simultaneous formulation is used, however, the equilibrium requirement need not be satisfied at intermediate optimization steps and some savings in computational effort might be expected from its absence. Unfortunately, the simultaneous method results in a much larger optimization problem, because the sizes of the constraint Jacobian and Lagrangian Hessian matrices are increased by the number of degrees of freedom of the structural problem. As a consequence, a dense projected Lagrangian implementation of this method, as in previous work,[7,8] requires much more storage and will almost always be less efficient than the nested approach. In this paper we describe the sparsity structure of the Jacobian and Hessian matrices associated with these two approaches and compare the performance of a sparse implementation of the simultaneous approach with that of the nested for optimization of geometrically nonlinear truss structures.

## II. Formulation

For simplicity, only truss-type structures subjected to a single-load case are considered. Multiple load conditions can be included in a way similar to that used in Ref. 19 for the linear case. Generalization to this case, to other sizing optimization problems, and other element types such as beams or plates is straightforward and will not change the basic structure of the Hessian and Jacobian matrices.

Presented as Paper 91-1093 at the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, Baltimore, MD, April 8-10, 1991; received June 11, 1991; revision received Nov. 25, 1991; accepted for publication Nov. 28, 1991. Copyright © 1991 by the American Institute of Aeronautics and Astronautics, Inc. All rights reserved.

*Graduate Assistant.

†Assistant Professor. Member AIAA.

### A. Nonlinear Truss Elements

The strain energy for a truss element $j$ with constant area $a_j$ and modulus of elasticity $E$ is

$$U_j = (1/2)Ea_jl_j\varepsilon_j^2 \tag{1}$$

where $l_j$ is the length of the element and $\varepsilon_j$ its longitudinal strain. For a three-dimensional truss element, $\varepsilon_j$ takes the form:

$$\varepsilon_j \equiv \frac{\delta l}{l} = \left\{1 + \frac{2}{l^2}[(y_4 - y_1)(u_4 - u_1) + (y_5 - y_2) \times (u_5 - u_2) + (y_6 - y_3)(u_6 - u_3)] + \frac{1}{l^2} [(u_4 - u_1)^2 + (u_5 - u_2)^2 + (u_6 - u_3)^2]\right\}^{1/2} - 1 \tag{2}$$

where the $y$'s represent element nodal coordinates and the $u$'s element nodal displacements.

Now, the total strain energy for the truss is

$$U = \sum_j^N \frac{1}{2} Ea_jl_j\varepsilon_j^2 \tag{3}$$

where $N$ is the total number of elements.

The $i$th element of the vector of (internal) nodal forces can be obtained using Castigliano's first theorem as:

$$P_i(\boldsymbol{a}, \boldsymbol{u}) = \frac{\partial U}{\partial u_i} = E \sum_j^N a_jl_j\varepsilon_j \frac{\partial \varepsilon_j}{\partial u_i} \tag{4}$$

where $\boldsymbol{a}$ and $\boldsymbol{u}$ are the vector of design areas and the vector of nodal displacements, respectively.

Only the case in which the external forces are constant is considered here. For a formulation that includes the variation of these forces with respect to the sizing variables, the interested reader should see Refs. 2 and 4.

Under the above circumstances the equations of equilibrium take the form:

$$\boldsymbol{\Psi}(\boldsymbol{a}, \boldsymbol{u}) \equiv \boldsymbol{P}(\boldsymbol{a}, \boldsymbol{u}) - \boldsymbol{F} = 0 \tag{5}$$

where $\boldsymbol{F}$ is the vector of externally applied loads.

If a Newton-type method is used to solve the nonlinear system of Eq. (5), its Jacobian will be needed. Because $\boldsymbol{F}$ is constant, the Jacobian of Eq. (5) with respect to the displacements is simply $\partial\boldsymbol{P}(\boldsymbol{a}, \boldsymbol{u})/\partial\boldsymbol{u}$, which is by definition the tangent stiffness matrix of the structure. Its components are obtained by taking the appropriate derivatives in Eq. (4). Each element $(i, k)$ of this matrix will then be

$$\frac{\partial P_i(\boldsymbol{a}, \boldsymbol{u})}{\partial u_k} = E \sum_j^N a_jl_j\left(\frac{\partial \varepsilon_j}{\partial u_i}\frac{\partial \varepsilon_j}{\partial u_k} + \varepsilon_j \frac{\partial^2 \varepsilon_j}{\partial u_i \partial u_k}\right) \tag{6}$$

Expression (6) was explicitly evaluated at the element level and the global tangent stiffness matrix assembled for the whole structure using the direct stiffness method.

### B. Nested Approach

Taking the volume as the objective function, the nonlinear mathematical programming problem for the nested case can be formulated as follows:

Minimize

$$V = \sum_{k=1}^{m} \left(b_k \sum_{j=1}^{J_k} l_j\right)$$

subject to:

$$\begin{aligned} g_{\sigma c} &\equiv \boldsymbol{\sigma} + \boldsymbol{\sigma}_c \geq 0 \\ g_{\sigma t} &\equiv \boldsymbol{\sigma}_t - \boldsymbol{\sigma} \geq 0 \\ g_{uL} &\equiv \boldsymbol{u} - \boldsymbol{u}_L \geq 0 \\ g_{uU} &\equiv \boldsymbol{u}_U - \boldsymbol{u} \geq 0 \\ \boldsymbol{b}_L &\leq \boldsymbol{b} \leq \boldsymbol{b}_U \end{aligned} \tag{7}$$

where: $m$ = number of design variables; $b_k$ = design area corresponding to element group $k$; $J_k$ = number of elements with area $b_k$; $\boldsymbol{\sigma}$ = vector of element stresses; $\boldsymbol{\sigma}_c$ = vector of allowable compressive stresses; $\boldsymbol{\sigma}_t$ = vector of allowable tensile stresses; $\boldsymbol{u}$ = vector of nodal displacements; $\boldsymbol{u}_L$ and $\boldsymbol{u}_U$ = vectors of displacement lower and upper bounds; $\boldsymbol{b}$ = vector of design variables (in this case, areas); and $\boldsymbol{b}_L$ and $\boldsymbol{b}_U$ = vectors with the values of lower and upper bounds of the design variables.

#### *1. Jacobian of the Stress Constraints*

If we now define $\boldsymbol{g}_\sigma$ as a column vector of dimension $2N$ with the values of the stress constraints, we can write the derivative of the constraints with respect to the design variables (i.e., the Jacobian matrix of the stress constraints) as:

$$\frac{\partial \boldsymbol{g}_\sigma}{\partial \boldsymbol{b}} = \begin{bmatrix} \dfrac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{b}} \\ -\dfrac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{b}} \end{bmatrix} \tag{8}$$

$\partial\boldsymbol{\sigma}/\partial\boldsymbol{b}$ may be found using the chain rule as:

$$\frac{d\boldsymbol{\sigma}}{d\boldsymbol{b}} = \frac{\partial \boldsymbol{\sigma}}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{b}} \tag{9}$$

Now, $\partial\sigma_j/\partial\boldsymbol{u} = E\partial\varepsilon_j/\partial\boldsymbol{u}$ and $\partial\varepsilon_j/\partial\boldsymbol{u}$ can be obtained from Eq. (2). It can be easily verified then that $\partial\boldsymbol{\sigma}/\partial\boldsymbol{u}$ is an $N \times n$ sparse matrix of direction cosines of the element axes multiplied by $E/l$. Each of its rows has at most six nonzero entries for space trusses and four for plane trusses. Here $l$ is the current length of the truss element and $n$ is the number of degrees of freedom after application of the appropriate boundary conditions.

The equations of equilibrium (5) must now be recast in terms of the more general variable $\boldsymbol{b}$.

$$\boldsymbol{\Psi}(\boldsymbol{b}, \boldsymbol{u}) \equiv \boldsymbol{P}(\boldsymbol{b}, \boldsymbol{u}) - \boldsymbol{F} = 0 \tag{10}$$

where we have retained the same symbols $\boldsymbol{\Psi}$ and $\boldsymbol{P}$ for the sake of simplicity.

To find $\partial\boldsymbol{u}/\partial\boldsymbol{b}$ we take the derivative of the right-hand side of Eq. (10) with respect to $\boldsymbol{b}$ to obtain:

$$\frac{\partial \boldsymbol{P}(\boldsymbol{b}, \boldsymbol{u})}{\partial \boldsymbol{b}} + \frac{\partial \boldsymbol{P}(\boldsymbol{b}, \boldsymbol{u})}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{b}} = 0 \tag{11}$$

$\partial\boldsymbol{P}/\partial\boldsymbol{b}$ is a sparse $n \times m$ matrix whose components are element nodal forces divided by the corresponding element sizes ($m$ is the number of element groups; i.e., the number of design variables). In the absence of element linking; i.e., when $m = N$, $\partial\boldsymbol{P}/\partial\boldsymbol{b}$ has the same sparsity structure as $(\partial\boldsymbol{\sigma}/\partial\boldsymbol{u})^T$ (six nonzero entries per row in three-dimensions and four nonzero entries in two dimensions, for truss elements). When $m < N$, the $k$th column of $\partial\boldsymbol{P}/\partial\boldsymbol{b}$ is a sum over the contributions of the elements in size group $k$. The matrix $\partial\boldsymbol{P}/\partial\boldsymbol{u}$ is as before the tangent stiffness matrix of the structure $\boldsymbol{K}_T$.

Equations (11) can now be solved for $\partial u/\partial b$, to obtain formally

$$\frac{\partial u}{\partial b} = -K_T^{-1}\frac{\partial P(b, u)}{\partial b} \tag{12}$$

and if this result is further substituted in (9) we get

$$\frac{\partial \sigma}{\partial b} = -\frac{\partial \sigma}{\partial u}K_T^{-1}\frac{\partial P(b, u)}{\partial b} \tag{13}$$

Depending on the order of multiplication of the matrices in (13), the procedures are referred to as the direct or adjoint methods in the literature[20,21] Given the characteristics of the problems presented here (fewer variables, in general, than constraints), the so-called direct approach is used. That is, $\partial u/\partial b$ is computed from Eq. (12) making use of the factored form of $K_T$ and the result substituted in Eq. (9).

Because the inverse of the tangent stiffness matrix is, in general, fully populated, the Jacobian of the stress constraints (8) will be dense.

### 2. *Jacobian of the Displacements Constraints*

In a manner analogous to that of Sect. II.B.1 the Jacobian of the displacement constraints can be written as:

$$\frac{\partial g_u}{\partial b} = \begin{bmatrix} \dfrac{\partial u}{\partial b} \\ -\dfrac{\partial u}{\partial b} \end{bmatrix} \tag{14}$$

where $g_u$ represents a vector with the values of all displacements constraints. $\partial u/\partial b$ is available from Eq. (12) and, as in the case of the stress Jacobian, it will be fully populated.

### 3. *Hessian of the Lagrangian Function*

If the optimization technique used involves reduced-gradient type methods with second order information for the search direction, or projected or augmented Lagrangian methods, it is necessary either to use the exact Hessian of the Lagrangian function if it is available, or to approximate it by one of the rank-one or rank-two update formulas. The BFGS update is one of the most popular.[7]

To examine the structure of this matrix, we first form the Lagrangian function corresponding to problem (7) and denote it by $L$

$$L(\lambda, b) = V - \hat{g}^T\lambda \tag{15}$$

where $\hat{g}$ is a vector with the values of all the *active* constraints including those that correspond to the lower and upper bounds and, $\lambda$ is the corresponding vector of Lagrange multipliers.

The gradient of Eq. (15) with respect to the areas is

$$\frac{\partial L(\lambda, b)}{\partial b} = \frac{\partial V}{\partial b} - \frac{\partial \hat{g}^T}{\partial b}\lambda \tag{16}$$

And the corresponding Hessian

$$H_n = G_V - \sum_j^t \lambda_j G_j \tag{17}$$

where $t$ = number of active constraints; $G_V$ = Hessian of the objective function, which in this case is equal to zero; $G_j$ = Hessian of constraint $j$; and, $\lambda_j$ = Lagrange multiplier corresponding to constraint $j$.

As in the case of the stress and displacement constraints, $H_n$ will be a dense matrix.

## C. Simultaneous Approach

In this case the equilibrium equations (5) are regarded as equality constraints and the variables of the nonlinear programming problem include the areas as well as the displacements of the structure.

The formal optimization problem may be posed as follows:

Minimize

$$V = \sum_{k=1}^{m}\left(b_k \sum_{j=1}^{J_k} l_j\right)$$

subject to:

$$P(b, u) - F = 0$$

$$g_{\sigma c} \equiv \sigma + \sigma_c \geq 0$$

$$g_{\sigma t} \equiv \sigma_t - \sigma \geq 0$$

$$x_L \leq x \leq x_U \tag{18}$$

where $m$ = number of design variables; $b_k$ = design area corresponding to group $k$ of elements; $J_k$ = number of elements with area $b_k$; $\sigma$ = vector of element stresses; $\sigma_c$ = vector of allowable compressive stresses; $\sigma_t$ = vector of allowable tensile stresses; $x_L$ = vector of lower bounds; $x_U$ = vector of upper bounds; $x^T \equiv [b^T \quad u^T]$; and the remaining symbols are as defined before.

It should be noted that the displacement constraints are represented here as side constraints.

### 1. *Jacobian of the Equilibrium Constraints*

The derivative of Eq. (5) with respect to $x$ yields

$$\frac{\partial \Psi(b, u)}{\partial x} = \begin{bmatrix} \dfrac{\partial P(b, u)}{\partial b} & \dfrac{\partial P(b, u)}{\partial u} \end{bmatrix} \tag{19}$$

The matrix $\partial P(b, u)/\partial b$ is the same as for the nested case and $\partial P(b, u)/\partial u$ is again the tangent stiffness matrix $K_T(b, u)$.

Therefore, the Jacobian of the equilibrium constraints is

$$\frac{\partial \Psi(b, u)}{\partial x} = \begin{bmatrix} \dfrac{\partial P(b, u)}{\partial b} & K_T(b, u) \end{bmatrix} \tag{20}$$

### 2. *Jacobian of the Stress Constraints*

Taking the derivative of $g_\sigma$ with respect to $x$ we get

$$\frac{\partial g_\sigma}{\partial x} = \begin{bmatrix} \dfrac{\partial g_\sigma}{\partial b} & \dfrac{\partial g_\sigma}{\partial u} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial \sigma}{\partial b} & \dfrac{\partial \sigma}{\partial u} \\ -\dfrac{\partial \sigma}{\partial b} & -\dfrac{\partial \sigma}{\partial u} \end{bmatrix} \tag{21}$$

In this case the stresses do not depend on the element sizes explicitly so: $\partial \sigma/\partial b = 0$. On the other hand, the matrix $\partial \sigma/\partial u$ is as described in section II.B.2 for the nested approach.

As a consequence, the Jacobian of the stress constraints can be written as:

$$\frac{\partial g_\sigma}{\partial x} = \begin{bmatrix} 0 & \dfrac{\partial \sigma}{\partial u} \\ 0 & -\dfrac{\partial \sigma}{\partial u} \end{bmatrix} \tag{22}$$

### 3. *Hessian of the Lagrangian Function*

The Lagrangian function corresponding to problem (18) may be written as:

$$L(\boldsymbol{\lambda}, \boldsymbol{x}) = V - \boldsymbol{\Psi}^T\boldsymbol{\lambda}_\Psi - \hat{\boldsymbol{g}}_{\sigma c}^T\boldsymbol{\lambda}_{gc} - \hat{\boldsymbol{g}}_{\sigma t}^T\boldsymbol{\lambda}_{gt} - \hat{\boldsymbol{c}}^T\boldsymbol{\lambda}_c \tag{23}$$

where $\boldsymbol{\lambda}_\Psi$ = Lagrange multipliers associated with the equilibrium constraints; $\hat{\boldsymbol{g}}_{\sigma c}$ = vector with the values of the active compressive stress constraints; $\boldsymbol{\lambda}_{gc}$ = Lagrange multipliers associated with the active compressive stress constraints; $\hat{\boldsymbol{g}}_{\sigma t}$ = vector with the values of the active tensile stress constraints; $\boldsymbol{\lambda}_{gt}$ = Lagrange multipliers associated with the active tensile stress constraints; $\hat{\boldsymbol{c}}$ = vector with the values of the active set of upper and lower bounds constraints; and $\boldsymbol{\lambda}_c$ = Lagrange multipliers associated with $\hat{\boldsymbol{c}}$.

Taking into account Eqs. (20) and (22), the gradient of Eq. (23) is then

$$\frac{\partial L}{\partial \boldsymbol{x}} = \frac{\partial V}{\partial \boldsymbol{x}} - \begin{bmatrix} \left(\dfrac{\partial \boldsymbol{P}}{\partial \boldsymbol{b}}\right)^T \boldsymbol{\lambda}_\Psi \\ \boldsymbol{K}_T\boldsymbol{\lambda}_\Psi \end{bmatrix} + \begin{bmatrix} 0 \\ \left(\dfrac{\partial \hat{\boldsymbol{\sigma}}}{\partial \boldsymbol{u}}\right)^T (\boldsymbol{\lambda}_{gt} - \boldsymbol{\lambda}_{gc}) \end{bmatrix} - \left(\frac{\partial \hat{\boldsymbol{c}}}{\partial \boldsymbol{x}}\right)^T \boldsymbol{\lambda}_c \tag{24}$$

The Hessian of the Lagrangian function is obtained as the derivative of Eq. (24)

$$\boldsymbol{H}_s = -\begin{bmatrix} 0 & \left(\dfrac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{b}}\boldsymbol{\lambda}_\Psi\right)^T \\ \dfrac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{b}}\boldsymbol{\lambda}_\Psi & \dfrac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{u}}\boldsymbol{\lambda}_\Psi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \dfrac{\partial^2 \boldsymbol{\sigma}}{\partial \boldsymbol{u}^2}(\boldsymbol{\lambda}_{gt} - \boldsymbol{\lambda}_{gc}) \end{bmatrix} \tag{25}$$

The nonzero structure of $(\partial \boldsymbol{K}_T/\partial \boldsymbol{b})\boldsymbol{\lambda}_\Psi$ is identical to that of $\partial \boldsymbol{P}/\partial \boldsymbol{b}$. The nonzero structure of $\partial^2\boldsymbol{\sigma}/\partial \boldsymbol{u}^2(\boldsymbol{\lambda}_{gt} - \boldsymbol{\lambda}_{gc})$ and $(\partial \boldsymbol{K}_T/\partial \boldsymbol{u})\boldsymbol{\lambda}_\Psi$ is more complex and depends on the number of elements incident to a given node of the structure. It can be verified, however, that each column of these two matrices, say that corresponding to global degree of freedom *i*, will in general contain contributions from all the elements that share degree of freedom *i*.

## III. Sparsity and Storage Characteristics

In this section we discuss the storage requirements of different algorithms for solving the optimization problem when projected Lagrangian methods are used. A detailed description of the techniques dealt with here may be found in Ref. 17.

In general, a solution to the following system representing first order optimality conditions is needed at each optimization step

$$\begin{bmatrix} \boldsymbol{H} & -\boldsymbol{A}^T \\ -\boldsymbol{A} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{g} \\ \boldsymbol{c} \end{bmatrix} \tag{26}$$

where $\boldsymbol{H}$ is the Hessian of the Lagrangian, $\boldsymbol{A}$ the Jacobian of the active constraints, $\boldsymbol{p}$ the search direction, $\boldsymbol{\lambda}$ the vector of Lagrange multipliers, $\boldsymbol{g}$ the gradient of the objective function, and $\boldsymbol{c}$ the value of the constraints at the current iterate.

The storage required for obtaining a solution to this system depends on the approach used. If a range space method is used, the system to be solved can be obtained from Eq. (26) by block Gaussian elimination

$$\boldsymbol{A}\boldsymbol{H}^{-1}\boldsymbol{A}^T\boldsymbol{\lambda} = -\boldsymbol{A}\boldsymbol{H}^{-1}\boldsymbol{g} - \boldsymbol{c} \tag{27}$$

Because the factors of $\boldsymbol{H}$ are required, fill-in may occur and the sparsity structure of $\boldsymbol{H}$ is not sufficient to predict the amount of memory needed. On the other hand, if a null space technique is used, the system to be solved takes the form

$$\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}\boldsymbol{p}_z = -\boldsymbol{Z}^T(\boldsymbol{g} + \boldsymbol{H}\boldsymbol{Y}\boldsymbol{p}_y) \tag{28}$$

where $\boldsymbol{Z}$ is a matrix whose columns form a basis for the null space of $\boldsymbol{A}$, $\boldsymbol{Y}$ is a matrix whose columns span the range space of $\boldsymbol{A}^T$ and $\boldsymbol{p}_y$ represent the range space portion of the search direction. The most storage efficient method to solve Eq. (28) is to compute $\boldsymbol{Z}$ using an orthogonal factorization of $\boldsymbol{A}$ and store only the information needed for constructing Householder matrices. If, in addition, a conjugate gradient algorithm is used, the storage requirements are limited to those of $\boldsymbol{H}$ and $\boldsymbol{A}$, because, in this case, it is not necessary to compute $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}$ explicitly (which will, in general be a dense matrix), but rather just the product $\boldsymbol{Z}^T\boldsymbol{H}\boldsymbol{Z}\boldsymbol{p}_z$.

To solve the complete system (26) iteratively by the conjugate residual method (the system is indefinite) constitutes a third alternative to the problem of finding the search direction $\boldsymbol{p}$. In this case, only the nonzeros of $\boldsymbol{H}$ and $\boldsymbol{A}$ are needed.

In the remainder of this section we present estimates of the number of nonzeros of the Hessian and Jacobian matrices for the simultaneous and nested approaches. Given the diversity of alternatives that are available to solve the optimization problem, estimates of the number of nonzero entries are necessarily related to the method used. We have assumed for comparison purposes a storage scheme consistent with an iterative solution of Eqs. (26). This may not be the most computationally efficient approach but it is certainly the most storage efficient one, because it requires only the nonzeros of the Jacobian and Hessian matrices. The following symbols are used in what follows: $n$ = number of degrees of freedom of the structure after applying boundary conditions; $m$ = number of groups of design areas; $d$ = number of degrees of freedom per element; $N$ = total number of elements; $e$ = maximum number of elements incident to a node; and $b$ = semibandwidth of tangent stiffness matrix.

### A. Nested Approach

According to the ideas presented in section II.B.1 the Jacobian matrix for the nested approach [see Eq. (8)] is a fully populated matrix and, therefore, has $Nm$ nonzero elements. Similarly, from Eq. (17) it is seen that the corresponding Hessian matrix $\boldsymbol{H}_n$ has $m^2$ nonzeros.

### B. Simultaneous Approach

#### 1. *Jacobian of the Equilibrium Constraints*

As noted before, $\partial \boldsymbol{P}/\partial \boldsymbol{b}$ is a sparse matrix composed of sums of internal nodal forces divided by elemental areas. Its *k*th column includes contributions from all the elements belonging to group *k*. In the absence of element linking or when the linked groups do not share any degrees of freedom, the number of nonzeros is $dN$. In all other cases $dN$ is an upper bound.

Now, an estimate for the upper bound of the number of nonzero entries of the tangent stiffness matrix will be the number of entries in the element stiffness matrices times the number of elements, that is: $d^2N$. If a direct band solver is used in the nested case, fill-in will demand the storage of the complete band, and the estimate would be $bn$. The estimate of $d^2N$ is valid, however, if an iterative method is used to solve the equilibrium equations according to our assumption.

#### 2. *Jacobian of the Stress Constraints*

As described in Section II.B.1 the structure of $\partial\boldsymbol{\sigma}/\partial \boldsymbol{u}$ is identical to that of $(\partial \boldsymbol{P}/\partial \boldsymbol{b})^T$ in the absence of linking. As a consequence, this matrix has also $dN$ nonzero elements.

### *3. Hessian of the Lagrangian Function*

All matrices in Eqs. (25) are sparse in nature. Upper bounds for their number of nonzero entries are as follows:

$$\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{b}} \boldsymbol{\lambda}_{\Psi} : dN$$

$$\frac{\partial^2 \boldsymbol{\sigma}}{\partial \boldsymbol{u}^2} (\boldsymbol{\lambda}_{gt} - \boldsymbol{\lambda}_{gc}) : n(e + 1)\ d/2$$

$$\frac{\partial \boldsymbol{K}_T}{\partial \boldsymbol{u}} \boldsymbol{\lambda}_{\Psi} : n(e + 1)\ d/2$$

The number of nonzeros of the main matrices for the simultaneous and nested approaches in the context of nonlinear analysis of reticular plane trusses is summarized in Table 1. The same estimates of number of nonzeroes, for structures of the type shown in Fig. 2, are plotted in Fig. 1. It is observed that the simultaneous method can be dramatically more efficient in terms of storage when an appropriate algorithm is used to solve the optimization problem. The crossover point at which the simultaneous method becomes more efficient is close to zero when there is no element linking and it moves to the right when a linking scheme is introduced. It can be concluded that for very large problems the simultaneous approach may be the only viable technique based on storage requirements alone.

**Table 1 Number of nonzero entries of main matrices**

| Matrix | Nested | Simultaneous |
|---|---|---|
| $\boldsymbol{K}_T$ | $d^2N$ | —— |
| $\partial\boldsymbol{\Psi}/\partial\boldsymbol{x}$ | —— | $(d^2N) + dN$ |
| $\partial\boldsymbol{g}_{\sigma}/\partial\boldsymbol{x}$ | $Nm$ | $dN$ |
| $\boldsymbol{H}$ | $m^2$ | $d(\partial N + n(e + 1)/2)$ |
| Total | $d^2N + Nm + m^2$ | $d(Nd + 4N + n(e + 1)/2)$ |

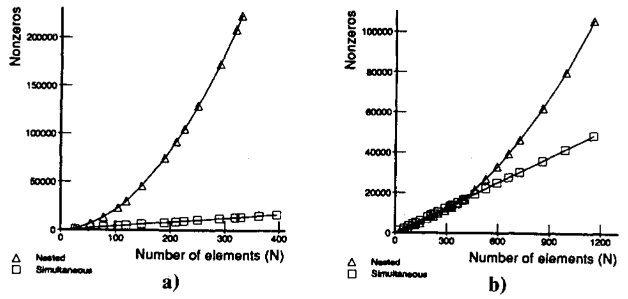

**Fig. 1 Number of nonzeros: a) no element linkages, b) element.**

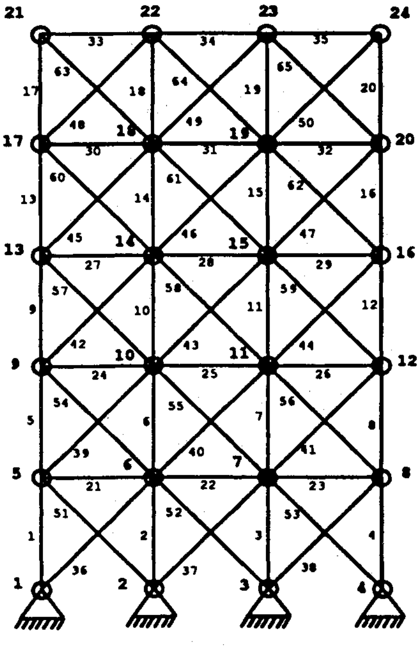

**Fig. 2 65-bar plane truss with 40 degrees of freedom.**

## IV. Solution Procedures

To demonstrate the efficacy of the simultaneous method relative to the nested, several structural optimization problems were solved using well-known and readily available projected Lagrangian codes: a sequential quadratic programming (SQP) algorithm available in the IMSL mathematical library and based on that developed by Schittkowski,[18] and the MINOS code for large scale problems.[22] The choice of these codes was motivated by the desire to solve each formulation as efficiently as possible. The nested formulation produces dense matrices and highly nonlinear constraints and, therefore, an SQP algorithm is most appropriate. The simultaneous formulation produces large, sparse matrices and mildly nonlinear constraints (inverse of stiffness matrix does not appear in the constraints). Therefore, a projected Lagrangian algorithm with a general subproblem, such as the one used by MINOS, should be more efficient.

The SQP routine used does not take into account the sparsity of either the Hessian or the Jacobian matrices. As a consequence, the quadratic programming subproblems associated with the simultaneous method are much larger and more expensive to solve than those of the nested case. On the other hand, the MINOS code uses a projected Lagrangian Method in conjunction with a general subproblem. It includes a number of features for large-scale, sparse problems. The Jacobian matrix is represented sparsely in terms of the nonzeros. A more efficient orthogonal, rather than orthonormal, null space basis is employed. A sparse LU factorization with a Markowitz ordering scheme and Bartels-Golub updates is used to construct the basis. Dense quasi-Newton updates are also used to approximate the projected Hessian. MINOS explicitly requires the values and positions of all the nonzero entries of the Jacobian matrix. Although the sparse nature of the Hessian is not exploited, the results obtained with MINOS provide useful information about the behavior of the simultaneous method when its sparse characteristics are considered. Furthermore, exploiting Hessian sparsity might not be beneficial because it will require the use of second-order information and a strategy for handling indefinite Hessians in the Newton algorithm.

## V. Numerical Results

Figure 2 illustrates the type of plane truss structures that were used for the numerical experiments. The number of bays and stories were adjusted to obtain different numbers of elements. No element linkages were assumed. The loads and the initial guesses for the areas were selected according to heuristics based on the number of bays and stories. The performance of the two codes compared is particularly sensitive to scaling. For the SQP solver, the objective function was multiplied by $10^3$ for the nested formulation and by $10^5$ for the simultaneous formulation. These scaling factors were found empirically to be "optimum". The same scaling factors were used with MINOS but, in addition, MINOS "SCALE ALL VARIABLES" option was utilized.

Although the optimization process was performed for geometrically nonlinear behavior, the *linear* displacements corresponding to the initial guesses of the areas were used as initial guesses in the simultaneous case. All numerical tests were performed on a SUN 4-260. Convergence to the same optimum was achieved in *all* cases tested. (Only stress constraints were active at the optimum for all the cases presented. When displacement constraints are likely to be active at the solution MINOS experiences serious convergence difficulties.)

Table 2 CPU time in seconds

| Structure | | | SQP | | MINOS | |
|---|---|---|---|---|---|---|
| Bays | Stories | Elements | Nested | Simultaneous | Nested | Simultaneous |
| 2 | 2 | 18 | 6 | 9 | 15 | 5 |
| 2 | 3 | 27 | 16 | 36 | 94 | 14 |
| 2 | 4 | 36 | 48 | 117 | 373 | 36 |
| 2 | 5 | 45 | 105 | 235 | 525 | 56 |
| 3 | 5 | 65 | 247 | 745 | 2783 | 144 |
| 3 | 6 | 78 | 309 | 1718 | 6477 | 255 |
| 3 | 7 | 91 | 471 | 3020 | 62195 | 428 |
| 3 | 8 | 104 | 1659 | 5298 | —— | 648 |
| 4 | 7 | 119 | 1925 | 6561 | [a] | 1003 |
| 4 | 8 | 136 | 2603 | 11056 | —— | 1755 |
| 4 | 9 | 153 | 3593 | 20055 | —— | 2179 |
| 4 | 10 | 170 | 3575 | 31537 | —— | 3333 |
| 5 | 9 | 189 | 5148 | 39110 | —— | 4802 |
| 5 | 10 | 210 | 7079 | 60186 | —— | 6545 |

[a] Stopped after 145,649 s of CPU time.

Table 3 65-bar two-dimensional truss

| Node | Fx, kg | Elem. | Init. area, cm$^2$ | Final area, cm$^2$ | Elem. | Init. area, cm$^2$ | Final area, cm$^2$ |
|---|---|---|---|---|---|---|---|
| 1 | 0.0 | 1 | 78.26 | 74.330600 | 33 | 7.00 | 7.000000 |
| 2 | 0.0 | 2 | 78.26 | 7.000000 | 34 | 7.00 | 7.000000 |
| 3 | 0.0 | 3 | 78.26 | 7.000000 | 35 | 7.00 | 7.000000 |
| 4 | 0.0 | 4 | 78.26 | 73.741402 | 36 | 7.00 | 49.908906 |
| 5 | 6063.0 | 5 | 56.00 | 59.870307 | 37 | 78.26 | 14.226423 |
| 6 | 6063.0 | 6 | 56.00 | 7.000000 | 38 | 78.26 | 11.895358 |
| 7 | 6063.0 | 7 | 56.00 | 7.000000 | 39 | 56.00 | 12.291351 |
| 8 | 6063.0 | 8 | 56.00 | 59.248159 | 40 | 56.00 | 39.465457 |
| 9 | 8103.0 | 9 | 36.37 | 41.975502 | 41 | 56.00 | 15.490034 |
| 10 | 8103.0 | 10 | 36.37 | 7.000000 | 42 | 36.37 | 14.393905 |
| 11 | 8103.0 | 11 | 36.37 | 7.000000 | 43 | 36.37 | 12.733977 |
| 12 | 8103.0 | 12 | 36.37 | 41.346744 | 44 | 36.37 | 28.255260 |
| 13 | 10270.0 | 13 | 19.80 | 12.182190 | 45 | 19.80 | 21.000440 |
| 14 | 10270.0 | 14 | 19.80 | 7.000000 | 46 | 19.80 | 10.634048 |
| 15 | 10270.0 | 15 | 19.80 | 7.000000 | 47 | 19.80 | 8.944051 |
| 16 | 10270.0 | 16 | 19.80 | 11.976571 | 48 | 7.00 | 7.000000 |
| 17 | 12550.0 | 17 | 7.00 | 7.000000 | 49 | 7.00 | 10.793100 |
| 18 | 12550.0 | 18 | 7.00 | 7.000000 | 50 | 7.00 | 7.000000 |
| 19 | 12550.0 | 19 | 7.00 | 7.000000 | 51 | 7.00 | 11.769546 |
| 20 | 12550.0 | 20 | 7.00 | 7.000000 | 52 | 78.26 | 14.352652 |
| 21 | 14930.0 | 21 | 7.00 | 7.000000 | 53 | 78.26 | 50.540894 |
| 22 | 14930.0 | 22 | 7.00 | 7.000000 | 54 | 56.00 | 15.376621 |
| 23 | 14930.0 | 23 | 7.00 | 7.000000 | 55 | 56.00 | 39.937279 |
| 24 | 14930.0 | 24 | 7.00 | 7.000000 | 56 | 56.00 | 12.299571 |
| | | 25 | 7.00 | 7.000000 | 57 | 36.37 | 28.600646 |
| | | 26 | 7.00 | 7.000000 | 58 | 36.37 | 12.644154 |
| | | 27 | 7.00 | 7.000000 | 59 | 36.37 | 14.400448 |
| | | 28 | 7.00 | 7.000000 | 60 | 19.80 | 8.882500 |
| | | 29 | 7.00 | 7.000000 | 61 | 19.80 | 10.609133 |
| | | 30 | 7.00 | 7.000000 | 62 | 19.80 | 20.750617 |
| | | 31 | 7.00 | 7.000000 | 63 | 7.00 | 7.000000 |
| | | 32 | 7.00 | 7.000000 | 64 | 7.00 | 10.679629 |
| | | | | | 65 | 7.00 | 7.000000 |

Table 2 shows the CPU times found for all the plane cases tested. In general, it can be observed that using MINOS in conjunction with the simultaneous formulation constitutes the most efficient approach. Close to the MINOS-simultaneous performance is the performance of the SQP using the nested formulation. The data corresponding to these two combinations are plotted in Fig. 3. It is observed that despite the fact that the simultaneous approach entails the solution of a much larger problem and the fact that no consideration is made of the sparsity of the Hessian matrix, the performance of MINOS is better than the performance of the SQP. The advantage of the simultaneous implementation is mainly due to the sparse representation of the Jacobian matrix of the constraints.

The most inefficient behavior is exhibited by MINOS with the nested formulation, as can be observed in Table 2 and in Fig. 4. This extremely inefficient performance is due to the fact that the number of calls to the gradient and function subroutines in MINOS is approximately one order of magnitude greater than for the SQP. The reason for this is that MINOS solves a general linearly constrained subproblem; whereas, SQP solves a much simpler quadratic subproblem. Because MINOS retains a linear approximation of the constraints for a sequence of quadratic approximations of the Lagrangian function, it is most efficient for linear or nearly linear constraints. In the nested case, the constraints are highly nonlinear in the design variables (they involve the inverse of the tangent stiffness matrix). This nonlinearity, which is relevant to the performance of MINOS, is not to be confused with the nonlinear behavior of the structure. Even for linearly behaving structures the constraints in the nested formulation are highly nonlinear in the design variables. In contrast, one of the effects of the simultaneous method is to reduce the nonlinearity in the constraints.

The CPU times for the SQP solver are plotted in Fig. 5. This figure illustrates the danger of a dense implementation of the simultaneous method. Because the sparsity of the Jacobian and Hessian matrices is not taken into account in this case, the performance of the nested method is superior to that of the simultaneous (for highly nonlinear problems the difference in performance will be less prominent).

It should be pointed out also that due to the presence of realistic stress constraints, the nonlinear behavior of the structures considered in this study was rather mild and each of the individual analysis required only a few Newton iterations to converge (typically, 4). It is expected that the greater the nonlinearity of the structure, the better the performance of the simultaneous formulation with respect to the nested.

For reproducibility purposes, the data and results for one of the two-dimensional cases are presented next. The external

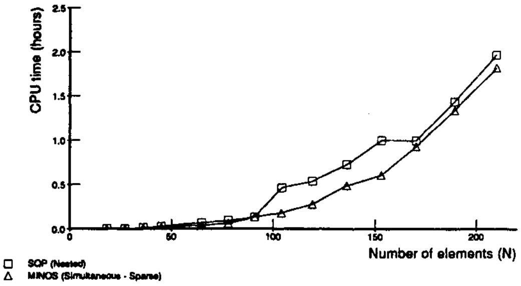

Fig. 3 MINOS-simultaneous vs SQP-nested.

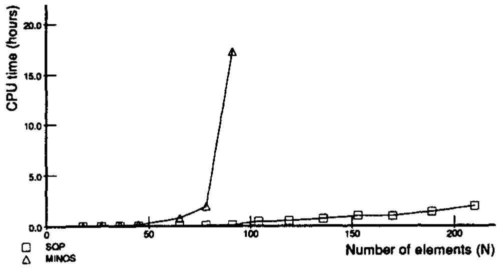

Fig. 4 Nested formulation. MINOS vs SQP solver.

Table 4 93-bar three-dimensional truss

| Node | Fx, kg | Fy, kg | Elem. | Init. area, cm$^2$ | Final area, cm$^2$ | Elem. | Init. area, cm$^2$ | Final area, cm$^2$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0000.0 | 0000.0 | 1 | 120.00 | 82.033283 | 47 | 95.00 | 10.000000 |
| 2 | 0000.0 | 0000.0 | 2 | 120.00 | 10.000000 | 48 | 95.00 | 10.000000 |
| 3 | 0000.0 | 0000.0 | 3 | 120.00 | 10.000000 | 49 | 95.00 | 17.689701 |
| 4 | 0000.0 | 0000.0 | 4 | 120.00 | 64.330464 | 50 | 95.00 | 10.000000 |
| 5 | 0000.0 | 0000.0 | 5 | 120.00 | 91.751513 | 51 | 95.00 | 57.887765 |
| 6 | 0000.0 | 0000.0 | 6 | 120.00 | 27.340036 | 52 | 95.00 | 29.388284 |
| 7 | 10000.0 | 10000.0 | 7 | 80.00 | 43.305745 | 53 | 95.00 | 23.095868 |
| 8 | 10000.0 | 10000.0 | 8 | 80.00 | 18.944379 | 54 | 10.00 | 10.000000 |
| 9 | 10000.0 | 10000.0 | 9 | 80.00 | 10.000000 | 55 | 10.00 | 10.000000 |
| 10 | 10000.0 | 10000.0 | 10 | 80.00 | 34.554565 | 56 | 10.00 | 10.000000 |
| 11 | 10000.0 | 10000.0 | 11 | 80.00 | 24.976876 | 57 | 120.00 | 10.000000 |
| 12 | 10000.0 | 10000.0 | 12 | 80.00 | 10.000000 | 58 | 120.00 | 13.396639 |
| 13 | 12000.0 | 12000.0 | 13 | 52.00 | 10.000000 | 59 | 120.00 | 30.916331 |
| 14 | 12000.0 | 12000.0 | 14 | 52.00 | 10.000000 | 60 | 120.00 | 10.000000 |
| 15 | 12000.0 | 12000.0 | 15 | 52.00 | 10.000000 | 61 | 80.00 | 10.141965 |
| 16 | 12000.0 | 12000.0 | 16 | 52.00 | 10.594795 | 62 | 80.00 | 23.099914 |
| 17 | 12000.0 | 12000.0 | 17 | 52.00 | 10.000000 | 63 | 80.00 | 17.028966 |
| 18 | 12000.0 | 12000.0 | 18 | 52.00 | 10.000000 | 64 | 80.00 | 18.281911 |
| 19 | 14500.0 | 14500.0 | 19 | 45.00 | 18.351643 | 65 | 80.00 | 16.520435 |
| 20 | 14500.0 | 14500.0 | 20 | 45.00 | 10.000000 | 66 | 52.00 | 10.000000 |
| 21 | 14500.0 | 14500.0 | 21 | 45.00 | 10.000000 | 67 | 52.00 | 22.148549 |
| 22 | 14500.0 | 14500.0 | 22 | 45.00 | 10.000000 | 68 | 52.00 | 10.856937 |
| 23 | 14500.0 | 14500.0 | 23 | 45.00 | 10.000000 | 69 | 52.00 | 10.000000 |
| 24 | 14500.0 | 14500.0 | 24 | 45.00 | 10.000000 | 70 | 52.00 | 23.467372 |
|  |  |  | 25 | 45.00 | 10.000000 | 71 | 45.00 | 26.407948 |
|  |  |  | 26 | 10.00 | 10.000000 | 72 | 45.00 | 10.000000 |
|  |  |  | 27 | 10.00 | 10.000000 | 73 | 45.00 | 10.000000 |
|  |  |  | 28 | 10.00 | 10.000000 | 74 | 45.00 | 10.000000 |
|  |  |  | 29 | 10.00 | 10.000000 | 75 | 45.00 | 10.000000 |
|  |  |  | 30 | 10.00 | 10.000000 | 76 | 10.00 | 10.000000 |
|  |  |  | 31 | 10.00 | 10.000000 | 77 | 10.00 | 10.000000 |
|  |  |  | 32 | 10.00 | 10.000000 | 78 | 10.00 | 11.114968 |
|  |  |  | 33 | 10.00 | 10.000000 | 79 | 10.00 | 10.000000 |
|  |  |  | 34 | 10.00 | 10.000000 | 80 | 10.00 | 10.000000 |
|  |  |  | 35 | 10.00 | 10.000000 | 81 | 10.00 | 10.000000 |
|  |  |  | 36 | 10.00 | 10.000000 | 82 | 120.00 | 10.312769 |
|  |  |  | 37 | 10.00 | 10.000000 | 83 | 120.00 | 10.000000 |
|  |  |  | 38 | 10.00 | 10.000000 | 84 | 120.00 | 10.000000 |
|  |  |  | 39 | 10.00 | 10.000000 | 85 | 120.00 | 10.910837 |
|  |  |  | 40 | 95.00 | 102.459900 | 86 | 80.00 | 10.000000 |
|  |  |  | 41 | 95.00 | 10.000000 | 87 | 80.00 | 10.000000 |
|  |  |  | 42 | 95.00 | 19.827895 | 88 | 80.00 | 12.617150 |
|  |  |  | 43 | 95.00 | 10.000000 | 89 | 80.00 | 10.000000 |
|  |  |  | 44 | 95.00 | 10.000000 | 90 | 80.00 | 10.000000 |
|  |  |  | 45 | 95.00 | 30.607032 | 91 | 52.00 | 10.000000 |
|  |  |  | 46 | 95.00 | 40.422882 | 92 | 52.00 | 10.000000 |
|  |  |  |  |  |  | 93 | 52.00 | 10.000000 |

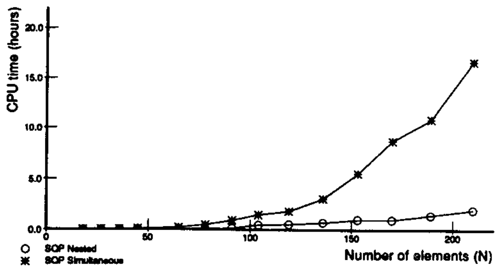

Fig. 5 SQP solver. Simultaneous vs nested.

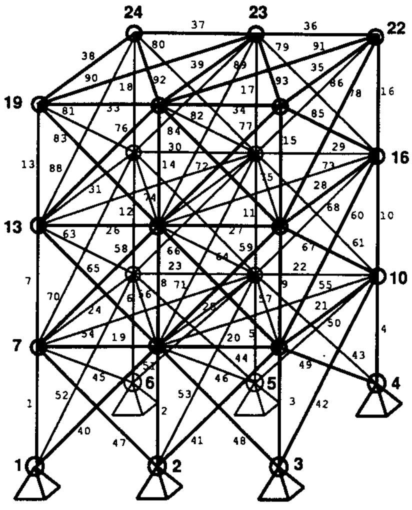

Fig. 6 Three-dimensional truss with 93 elements and 58 degrees of freedom.

forces and the individual element sizes are shown in Table 3. The numbering scheme corresponds to that in Fig. 2.

Main data are:

Three bays, five stories, 65 elements, 24 nodes.

Dimensions: vertical bars: 3.0 m, horizontal bars: 4.0 m, diagonal bars: 5.0 m

Material: Aluminum ⇛ $E = 7.0 \times 10^5$ kg/cm$^2$.

Allowable compressible stress: 1,700 kg/cm$^2$.

Allowable tensile stress: 1,700 kg/cm$^2$.

Method of solution: MINOS, simultaneous

Degrees of freedom: 40; optimization variables: 105.

Number of constraints: 170

Final volume: 444,039 cm$^3$; CPU time: 144 s

For purposes of completeness a three-dimensional case was included in the numerical experiments. The corresponding truss structure is illustrated in Fig. 6. The CPU times found for this case are the following:

SQP-nested: 372 s

SQP-simultaneous: 2,695 s

MINOS-nested: 15,439 s

MINOS-simultaneous: 490 s

The better performance of the SQP in this case is attributed to the rather mild nonlinearity of the structural problem and to the differences in implementation between MINOS and the SQP solver mentioned above.

The general data for the three-dimensional truss structure are given next. Element sizes and external forces are presented in Table 4. The numbering scheme is the one indicated in Fig. 6.

Main data are:

Three stories, 93 elements, 24 nodes

Support conditions: All degrees of freedom restricted except for nodes 2, 3, 4, and 5, which are free to move in the $x$ direction.

Dimensions: vertical bars: 3.0 m, horizontal bars: 4.0 m, diagonal bars: 5.0 m

Material: Aluminum ⇛ $E = 7.0 \times 10^5$ kg/cm$^2$

Allowable compressible stress: 1,700 kg/cm$^2$

Allowable tensile stress: 1,700 kg/cm$^2$

Method of solution: SQP, nested

Degrees of freedom: 58; optimization variables: 151

Number of constraints: 186

Final volume: 685,873 cm$^3$; CPU time: 372 s

## VI. Conclusions

Two different approaches to the problem of geometrically nonlinear structural optimization have been reexamined. The traditional nested approach requires the solution of the system of equilibrium equations at each optimization iteration. In contrast, the simultaneous approach regards the equilibrium equations as equality constraints and the displacements of the structure as design variables, allowing simultaneous analysis and design. For these two methods, the basic structure of the Jacobian and Hessian matrices in the context of volume minimization of geometrically nonlinear behavior of truss structures has been presented. For the class of problems considered, it has been shown that exploitation of the sparsity characteristics of its related matrices is critical to the success of the simultaneous approach, in terms of both computational and storage efficiency. In addition, our results indicate that a sparse implementation of an SQP algorithm may be the most efficient technique for the solution of large-scale structural optimization problems, provided that the cost of producing a different sparse basis at each iteration does not overwhelm the superior algorithmic performance of SQP.

## Acknowledgments

This research was partially supported by the Engineering Design Research Center, an NSF Engineering Design Center at Carnegie Mellon University, and NSF Grant DDM-9009 597. The authors gratefully acknowledge the valuable comments of the reviewers.

## References

[1] Haftka, R. T., "Integrated Nonlinear Analysis and Design," *AIAA Journal*, Vol. 27, No. 11, 1989, pp. 1622–1627

[2] Ryu, Y. S., Haririan, M., Wu, C. C., and Arora, J. S., "Structural Design Sensitivity of Nonlinear Response," *Computers & Structures*, Vol. 21, No. 1/2, 1985, pp. 245–255.

[3] Wu, C. C., and Arora, J. S., "Simultaneous Analysis and Design of Nonlinear Response," *Engineering with Computers*, Vol. 2, No. 1, 1987, pp. 53–63.

[4] Wu, C. C., and Arora, J. S., "Design Sensitivity Analysis of Nonlinear Response using Incremental Procedure," *AIAA Journal*, Vol. 25, No. 8, 1987, pp. 1118–1125.

[5] Schmit, L. A., and Fox, L. R., "An Integrated Approach to Structural Synthesis and Analysis," *AIAA Journal*, Vol. 3, No. 6, 1965, pp. 1104–1112.

[6] Fox, L. R., and Schmit, L. A., "Advances in the Integrated Approach to Structural Synthesis," *Journal of Spacecraft and Rockets*, Vol. 3, No. 6, 1966, pp. 858–866.

[7] Haftka, R. T., "Simultaneous Analysis and Design," *AIAA Journal*, Vol. 23, No. 7, 1985, pp. 1099–1103.

[8] Haftka, R. T., and Kamat, M. P., "Simultaneous Nonlinear Structural Analysis and Design," *Computational Mechanics*, Vol. 4, No. 6, 1989, pp. 409–416.

[9] Cuthrell, J. E., and Biegler, L. T., "Simultaneous Solution and Optimization of Process Flowsheets with Differential Equation Models," *Chemical Engineering Research & Design*, Vol. 64, No. 9, 1986, p. 341.

[10] Biegler, L. T., "On the Simultaneous Solution and Optimization of Large Scale Engineering Systems," *Computers & Chemical Engineering*, Vol. 12, No. 5, 1988, pp. 357–369.

[11] Hrymak, A. N., McRae, G. J., and Westerberg, A. W., "Combined Analysis and Optimization of Extended Heat Transfer Surfaces," *Journal of Heat Transfer*, Vol. 107, No. 3, 1985, pp. 527–532.

[12] Chao, N. H., Fenves, S. J., and Westerberg, A. W., "Application of a Reduced Quadratic Programming Technique to Optimal Struc-

tural Design," *New Directions in Optimum Structural Design*, edited by A. Atrek, R. H. Gallagher, K. M. Ragsdell, and O. C. Zienkiewicz, Wiley, New York, 1984, pp. 413–427.

[13] Grossmann, I. E., "Mixed-Integer Nonlinear Programming Techniques for the Synthesis of Engineering Systems," *Research in Engineering Design*, Vol. 1, No. 3/4, 1990, pp. 205–228.

[14] Smaoui, H., and Schmit, L. A., "An Integrated Approach to the Synthesis of Geometrically Nonlinear Structures," *International Journal for Numerical Methods in Engineering*, Vol. 26, No. 3, 1988, pp. 555–570.

[15] Bendsøe, M. P., Ben-Tal, A., and Haftka, R., "New Displacement-Based Methods for Optimal Truss Topology Design," *Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd SDM Conference*, Baltimore, MD, April 1991.

[16] Ghattas, O. N., and Schrader, S. K., "A Nonlinear Programming Formulation of the Topological Structural Optimization Problem," submitted for publication.

[17] Gill, P. E., Murray, W., and Wright, M. H., *Practical Optimization*, Academic Press, New York, 1981.

[18] Schittkowski, K., "NLPQL: A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems," *Annals of Operations Research*, Vol. 5, No. 6, 1985, pp. 485–500.

[19] Wakabayashi, T., "Computational Efficiency and Storage Requirements for the Hierarchical and Simultaneous Methods in Structural Optimization," Master's Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, 1990.

[20] Haftka, R. T., Gurdal, Z., and Kamat, M. P., *Elements of Structural Optimization*, Kluwer Academic Publishers, The Netherlands, 1990.

[21] Haug, E. J., Choi, K. K., and Komov, V., *Design Sensitivity Analysis of Structural Systems*, Academic Press, New York, 1986.

[22] Murtagh, B. A., and Sauders, M. A. "A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints," *Math. Prog. Study on Constrained Optimization*, Rept. SOL 80-1R, Dept. of Operations Research, Stanford University, Stanford, CA, 1980.