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

Omar N. Ghattas, Carlos E. Orozco

EDRC 12-43-91

A Sparse Approach to Simultaneous Analysis and Design of Geometrically Nonlinear Structures[1]

Omar N. Ghattas Carlos E. Orozco
Department of Civil Engineering
Carnegie Mellon University

July 19, 1991

[1]This work has been supported by the Engineering Design Research Center, an NSF Engineering Research Center.

Abstract

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 of the methods that has been 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.

Chapter 1

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 which may be assigned either 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 now 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 [3, 4, 5, 6]. The nonlinear case is particularly amenable to such an improvement since it is itself an iterative process. Wu and Arora [5], for instance, use the sensitivity information of the behavior of the structure to provide a better initial guess for the next nonlinear analysis. Haftka [3] 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. Within the structural context this alternative has been proposed by Haftka [1, 2]; the idea has also been used successfully by Biegler *et al* [12] in the chemical engineering context. The resulting approach is usually termed simultaneous.

Although there are several procedures to solve nonlinear mathematical programming problems, it is now generally accepted that projected Lagrangian methods provide the most efficient algorithms to solve this class of problems [9, 7]. 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 since 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, since 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 [1, 2], 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.

Chapter 2

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 [11] 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.

2.1 Nonlinear truss elements

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

$$Uj = \tfrac{1}{2} Eajljej \tag{2.1}$$

where lj is the length of the element and e_3 its longitudinal strain. For a three dimensional truss element, Sj takes the form:

$$Sj = \tfrac{1}{2} Eaj\{[(y_4 - y_1)(u_4 - u_1) + (y_5 - y_2)(u_5 - u_2) + (y_6 - y_3)(u_6 - u_3)] + \tfrac{1}{2}[(u_4 - u_1)^2 + (u_4 - u_1)^2 + (u_4 - u_1)^2]\}^{1/2} - 1 \tag{2.2}$$

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

Now, the total strain energy for the truss is:

$$U = \sum_{j}^{N} \tfrac{1}{2} Eajlje_j^2 \tag{2.3}$$

where N is the total number of elements.

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

$$P_i(\mathbf{a},\mathbf{u}) = \mathbf{g} \quad \mathbf{a} \qquad u_i \tag{2.4}$$

where a and 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 references [4] and [6].

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

$$*(\mathbf{a},\mathbf{u}) = \mathbf{P}(\mathbf{a},\mathbf{u}) - \mathbf{F} = \mathbf{0} \tag{2.5}$$

where F is the vector of externally applied loads.

If a Newton-type method is used to solve the nonlinear system of equations (2.5), its Jacobian will be needed. Since F is constant, the Jacobian of (2.5) with respect to the displacements is simply #P(a, u)/#u, which is by definition the tangent stiffness matrix of the structure. Its components are obtained by taking the appropriate derivatives in (2.4) . Each element (i, *k)* of this matrix will then be:

$$\frac{\partial P_i(\mathbf{a},\mathbf{u})}{du_k} = E\sum_j^N a_j l_j \left(\frac{\partial \varepsilon_j}{\partial u_i}\frac{\partial \varepsilon_j}{\partial u_k} + \varepsilon_j \frac{\partial^2 \varepsilon_j}{\partial u_i \partial u_k}\right) \tag{2.6}$$

Expression (2.6) was explicitly evaluated at the element level and the global tangent stiffness matrix assembled for the whole structure using the direct stiffness method.

2.2 Nested approach

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

$$\begin{aligned}
&\textit{Minimize } V = \sum_{k=1}^{TO} \left(\&c \sum_{j=1}^{Jk} 5Z\ \Lambda'\right) \\
&\textit{subject to :} \\
&g_{<7C} = <7 + <7_c \geq 0 \\
&g_{<7t} = <7_t - <7 \geq 0 \\
&g_{uL} = U - UL \geq 0 \\
&g_u u = uu - u \geq 0 \\
&bL \leq b \leq bjj
\end{aligned} \tag{2.7}$$

where:

m : Number of design variables.

bk : Design area corresponding to element group fc.

Jk : Number of elements with area 6*.

a : Vector of element stresses.

cr_c : Vector of allowable compressive stresses.

at : Vector of allowable tensile stresses.

u : Vector of nodal displacements.

UE, and *uu* are vectors of displacement lower and upper bounds.

b : Vector of design variables. (In this case areas).

bf, and *bu* are vectors with the values of lower and upper bounds of the design variables.

Jacobian of the stress constraints

If we now define g^ 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:

$$\frac{\partial g_\sigma}{} - \mathrm{I}^{\Gamma} \quad \% \quad \mathrm{I}^{\urcorner} \qquad (2\ 81$$

dafdh may be found using the chain rule as:

$$\frac{\partial \sigma}{db} \quad \frac{\partial \sigma}{0\,u\,0\,b}\frac{\partial u}{} \qquad > \qquad (2.9)$$

Now, *d<Tj/du = Edsj/dvi* and *d€j/du* can be obtained from Eqn. (2.2). It can be easily verified then that *da/du* is an *N* x n sparse matrix of direction cosines of the element axes multiplied by *E/l.* Each of its rows has at most 6 nonzero entries for space trusses and 4 for plane trusses. Here / is the initial 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 (2.5) must now be recast in terms of the more general variable b:

$$*(b, u) \equiv P(b, u) - F = 0 \qquad (2.10)$$

where we have retained the same symbols ^ and P for the sake of simplicity.

To find *dn/db* we take the derivative of the right hand side of (2.10) with respect to b to obtain:

$$\frac{ap(bu)}{\partial b} + \frac{ap(bu)}{\partial u}\frac{gu}{5\,b} = 0 \qquad (2.11)$$

#P/c?b is a sparse *n* x *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* = JV, *dP/db* has the same sparsity structure as $(da/d\backslash i)^T$ (6 nonzero entries per row in 3D and 4 nonzero entries in 2D, for truss elements). When *m* < *N,* the fcth column of *dP/db* is a sum over the contributions of the elements in size group *k.* The matrix *dPjdu* is as before the tangent stiffness matrix of the structure Kj.

Equations (2.11) can now be solved for du/db, to obtain formally:

$$\frac{}{db} = -\mathbf{K}_T^{-1} \frac{\partial \mathbf{P}(\mathbf{b}, \mathbf{u})}{\partial \mathbf{b}} \tag{2.12}$$

and if this result is further substituted in (2.9) we get:

$$\frac{d^*}{db} \sim \text{"}\bar{\mathbf{a}} \wedge \mathbf{K}_r \frac{dP(b,u)}{db} \tag{2\_13}$$

Depending on the order of multiplication of the matrices in (2.13), the procedures are referred to as the direct or adjoint methods in the literature [8,10]. Given the characteristics of the problems presented here (fewer variables, in general, than constraints), the so-called direct approach is used. That is, *dn/db* is computed from equation (2.12) making use of the factored form of Ky and the result substituted in (2.9).

Because the inverse of the tangent stiffness matrix is, in general, fully populated, the Jacobian of the stress constraints (2.8) will be dense.

Jacobian of the displacements constraints

In a manner analogous to that of section 2.2 the Jacobian of the displacement constraints can be written as:

$$\frac{\partial \mathbf{g}_u}{} = \begin{bmatrix} \frac{\partial \mathbf{u}}{\partial \mathbf{b}} \\ -\frac{\partial \mathbf{u}}{\partial \mathbf{b}} \end{bmatrix} \tag{2.14}$$

where g_u represents a vector with the values of all displacements constraints, *dyi/db* is available from (2.12) and, as in the case of the stress Jacobian, it will be fully populated.

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 [9].

To examine the structure of this matrix, we first form the Lagrangian function corresponding to problem (2.7) and denote it by *L:*

$$L(\boldsymbol{\lambda}, \mathbf{b}) = \mathrm{V} - \mathrm{g}^T \mathrm{A} \tag{2.15}$$

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,

A is the corresponding vector of Lagrange multipliers.

The gradient of (2.15) with respect to the areas is:

$$\frac{dL(X,b)}{db} = \frac{dV}{db} - \lambda^T \frac{d\hat{g}}{db} \tag{2.16}$$

And the corresponding Hessian:

$$H_n = G_V - \sum_i \lambda_j G_i \tag{2.17}$$

where:

t : Number of active constraints.

Gy • Hessian of the objective function which in this case is equal to zero.

Gj : Hessian of constraint *j*.

\j *:* Lagrange multiplier corresponding to constraint *j*

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

2.3 Simultaneous approach

In this case the equilibrium equations (2.5) 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:

$$\begin{aligned} &\textit{Minimize } V = \sum_{k=1}^{m} \left(b_k \sum_{j=1}^{J_k} l_j \right) \\ &\textit{subject to :} \\ &P(b,u) - F = 0 \\ &g_{ac} \equiv a + < r_c \geq 0 \\ &g_{at} \equiv j_t - < r \geq 0 \\ &XL \leq X \leq XU \end{aligned} \tag{2.18}$$

where:

m : Number of design variables.

bk .' Design area corresponding to group k of elements.

Jk : Number of elements with area 6*.

a : Vector of element stresses.

cr_c : Vector of allowable compressive stresses.

$<r_t$: Vector of allowable tensile stresses.

XL .* Vector of lower bounds.

xu : Vector of upper bounds.

$x^T \equiv [\, b^r \;\; u^T \,]$ and,

the remaining symbols are as defined before.

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

Jacobian of the equilibrium constraints

The derivative of (2.5) with respect to x yields:

$$\frac{\partial \Psi(\mathbf{b},\mathbf{u})}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{P}(\mathbf{b},\mathbf{u})}{\partial \mathbf{b}} \quad \frac{\partial \mathbf{P}(\mathbf{b},\mathbf{u})}{\partial \mathbf{u}} \right] \quad (2.19)$$

The matrix #P(b, u)/#b is the same as for the nested case and #P(b, \i)/d\i is again the tangent stiffness matrix Kx(b,u).

Therefore, the Jacobian of the equilibrium constraints is:

$$\frac{\partial \Psi(\mathbf{b},\mathbf{u})}{\partial \mathbf{x}} = \left[\frac{\partial \mathbf{P}(\mathbf{b},\mathbf{u})}{\partial \mathbf{b}} \quad \mathbf{K}_T(\mathbf{b},\mathbf{u}) \right] \quad (2.20)$$

Jacobian of the stress constraints

Taking the derivative of g^ with respect to x we get:

$$\frac{\partial \mathbf{g}_\sigma}{\partial \mathbf{x}} - \left[\frac{\partial \mathbf{g}_\sigma}{\partial \mathbf{b}} \quad \frac{\partial \mathbf{g}_\sigma}{\partial \mathbf{u}} \right] = \begin{bmatrix} \frac{\partial \sigma}{\partial \mathbf{b}} & \frac{\partial \sigma}{\partial \mathbf{u}} \\ -\frac{\partial \sigma}{\partial \mathbf{b}} & -\frac{\partial \sigma}{\partial \mathbf{u}} \end{bmatrix} \quad (2.21)$$

In this case the stresses do not depend explicitly on the element sizes so: $dcr/db = 0$. On the other hand, the matrix da/du is as described in section 2.2 for the nested approach.

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

$$\frac{\partial \mathbf{g}_\sigma}{\partial \mathbf{x}} = \begin{bmatrix} \mathbf{0} & \frac{SSL}{\partial u} \\ \mathbf{0} & -\frac{}{\partial u} \end{bmatrix} \quad (2.22)$$

Hessian of the Lagrangian Function

The Lagrangian function corresponding to problem (2.18) may be written as:

$$L(\mathrm{A}, \mathrm{x}) = V - *^{\mathrm{T}}\mathrm{A}* - \hat{\mathbf{g}}\mathbf{J}_c\mathbf{A}_{yc} - \hat{\mathrm{g}}\pounds X_g t - \hat{c}^T X_c \qquad (2.23)$$

where:

$\backslash y$: Lagrange multipliers associated with the equilibrium constraints.

$\hat{g}$<ıc • Vector with the values of the active compressive stress constraints.

Xg_C : Lagrange multipliers associated with the active compressive stress constraints.

$\hat{g}_{at}$: Vector with the values of the active tensile stress constraints.

$X_g t$: Lagrange multipliers associated with the active tensile stress constraints.

$\hat{\mathbf{c}}$: Vector with the values of the active set of upper and lower bounds constraints.

$\mathrm{A_c}$: Lagrange multipliers associated with $\hat{\mathbf{c}}$

Taking into account (2.20) and (2.22) the gradient of (2.23) is then:

$$\frac{\partial L}{\partial \mathbf{x}} = \frac{\partial V}{\partial \mathbf{x}} - \begin{bmatrix} \left(\frac{\partial \mathbf{P}}{\partial \mathbf{b}}\right)^T \lambda_\Psi \\ \mathbf{K}_T \lambda_\Psi \end{bmatrix} + \begin{bmatrix} 0 \\ \left(\frac{\partial \hat{\sigma}}{\partial \mathbf{u}}\right)^T (\lambda_{gt} - \lambda_{gc}) \end{bmatrix} - \{\}\pounds)^{\left(\frac{\partial \hat{c}}{\partial}\right)^T} X_C \qquad (2.24)$$

The Hessian of the Lagrangian function is obtained as the derivative of (2.24) :

$$\mathrm{H} - \begin{bmatrix} 0 & \left(\frac{\partial \mathbf{K}_T}{\partial \mathbf{b}} \lambda_\Psi\right)^T \\ \frac{\partial \mathbf{K}_T}{\partial \mathbf{b}} \lambda_\Psi & \frac{\partial \mathbf{K}_T}{\partial \mathbf{u}} \lambda_\Psi \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \frac{\partial^2 \sigma}{\partial \mathbf{u}^2} (\lambda_{gt} - \lambda_{gc}) \end{bmatrix} \qquad (2.25)$$

The nonzero structure of $\wedge \mathrm{f}\,\mathrm{A}*$ is identical to that of ***dP/db.*** The nonzero structure of $d^2 < T/du^2 (X_g t - \mathrm{A}_{5C})$ and $\wedge\wedge$-Xy 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.

Chapter 3

Sparsity Characteristics

In this section we present estimates of the number of nonzeros for the Hessian and Jacobian matrices of both approaches. The following symbols will be used :

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.

6 : Semibandwidth of tangent stiffness matrix including diagonal elements.

3.1 Jacobian, nested approach

According to the ideas presented in section 2.2 the matrix of equation (2.8) is a fully populated matrix and therefore it has *Nm* nonzero elements.

3.2 Hessian, nested approach

Similarly, from (2.17) it is seen that H_n has m^2 nonzeros.

3*3 Simultaneous Approach

Jacobian of the equilibrium constraints

As noted before, #P/#b is a sparse matrix composed of sums of internal nodal forces divided by elemental areas. Its *kth.* 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.

Table 3.1: Number of nonzero entries of main matrices

Matrix	Nested	Simultaneous
K_T	d^2N	-
dV/dx	-	$(d?N)+dN$
$dgjdx$	Nm	dN
H	m^2	$_Ld\{2N + n(e+l)/2)$
TOTAL	$d?N+Nm+m^2$	$d(Nd+4N + n(e + l)/2)$

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: o^iV.[1]

Jacobian of the stress constraints

As described in section 2.2 the structure of $d<r/du$ is identical to that of $(dP/db)^T$ in the absence of linking. As a consequence, this matrix has also dN nonzero elements.

Hessian of the Lagrangian function

All matrices in (2.25) are sparse in nature. Upper bounds for their number of nonzero entries are as follows:

$\frac{\partial K_T}{\partial b}\lambda_\Psi$: dN.

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

$*f_xA*$: $n(e+1)d/2$.

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 3.1.

3.4 Storage required for the solution of the optimization problem

In this section we relate the nonzero structure of the Jacobian and Hessian matrices with the storage that would be required for the solution of the optimization problem when projected Lagrangian methods are used. A detailed description of the techniques dealt with here may be found in [9].

[a] If this matrix is stored in band form the estimate would be 6n, but since this matrix is common to both approaches the above bound was used for comparison purposes.

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

$$\begin{bmatrix} \mathbf{H} & -\mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} -\mathbf{g} \\ \mathbf{c} \end{bmatrix} \tag{3.1}$$

where H is the Hessian of the Lagrangian, A the Jacobian of the active constraints, p the search direction, A the vector of Lagrange multipliers, g the gradient of the objective function and 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 (3.1) by block Gaussian elimination:

$$AH^{-1}A^TA = -\mathbf{AIT}\hat{}\mathbf{g} - \mathbf{c} \tag{3.2}$$

Since the factors of H are required, fill-in may occur and the sparsity structure of 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:

$$Z^THZ_{Pz} = -Z^r(g + \mathbf{HY}\mathbf{p}_y) \tag{3.3}$$

where Z is a matrix whose columns form a basis for the null space of A, Y is a matrix whose columns span the range space of A and p_y represents the range space portion of the search direction. The most storage efficient method to solve (3.3) is to compute Z using an orthogonal factorization of 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 H and A, since in this case it is not necessary to compute $\mathbf{Z}^r\mathbf{HZ}$ explicitly (which will, in general be dense) but rather just the product $\mathbf{Z}^T\mathbf{HZp}_2$.

To solve the complete system (3.1) iteratively by the conjugate residual method (the system is indefinite) constitutes a third alternative to the problem of finding the search direction p. In this case, only the nonzeros of H and A are needed.

To illustrate the different storage characteristics of the simultaneous and nested approaches, the number of nonzero entries of the main matrices of these two methods, for structures of the type shown in Figure 5.1, is plotted in Figure 3.1. 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.

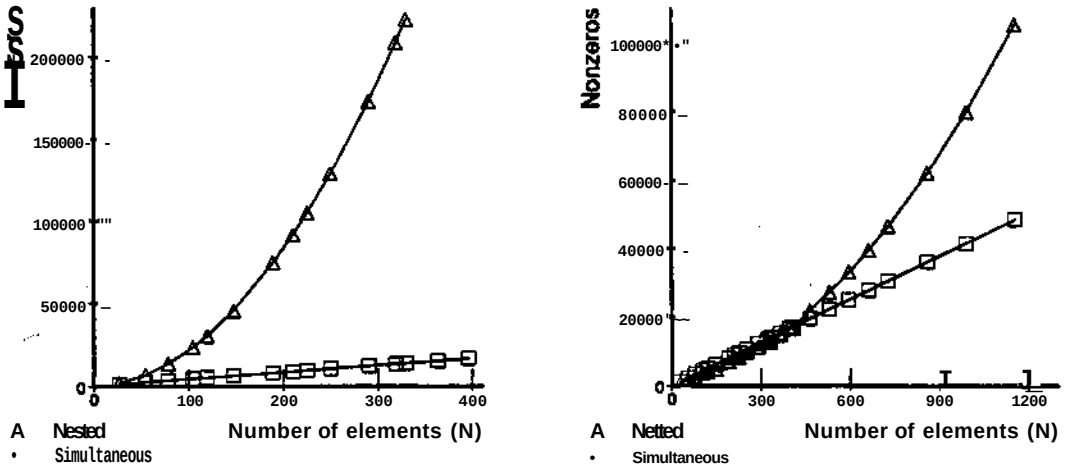

Figure 3.1: Number of nonzeros : a) No element linkages, b) Element linkages

Chapter 4

Solution procedures

The optimization problems (2.7) and (2.18) were first solved using a sequential quadratic programming (SQP) algorithm available in the IMSL mathematical library and based on that developed by Schittkowski [7]. The same set of problems was then solved with the MINOS code for large scale problems [13].

Schittkowsky's SQP routine 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 still provide useful information about the behavior of the simultaneous method when its sparse characteristics are considered.

Chapter 5

Numerical results

Figure 5.1 illustrates the type of truss structures that were used for the numerical experiments. The number of bays and stories were adjusted to obtain different numbers of elements. The loads and the initial guesses for the areas were selected according to heuristics based on the number of bays and stories. 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.

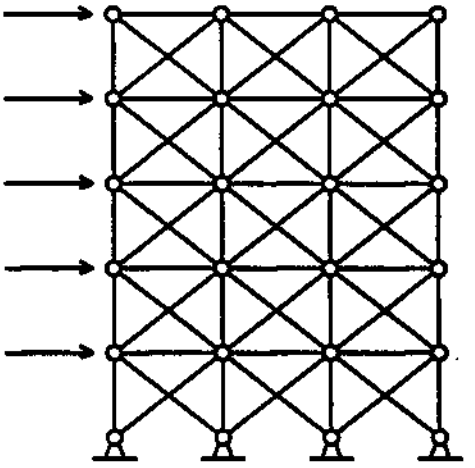

Figure 5.1: Example of plane truss used for the numerical experiments.

Figure 5.2 shows the results obtained using the SQP solver versus the results obtained using MINOS for the nested case (with no element linkages). It can be observed that in this case MINOS is extremely inefficient as compared to the SQP due mainly 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. Since 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.

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. It is anticipated that for highly nonlinear structural behavior the difference in performance will be even greater.

Figure 5.3 illustrates the danger of a dense implementation of the simultaneous method. Since 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).

Figure 5.4 shows the performance of MINOS for the nested and simultaneous formulations. The clear advantage of the simultaneous implementation is mainly due to the sparse representation of the Jacobian matrix of the constraints.

The results for the best performing code for each of the two approaches (simultaneous and nested) are presented in Figure 5.5. It should be 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. Although this superiority is small, it should be noted that it will increase as the nonlinearities in the structure become more prominent.

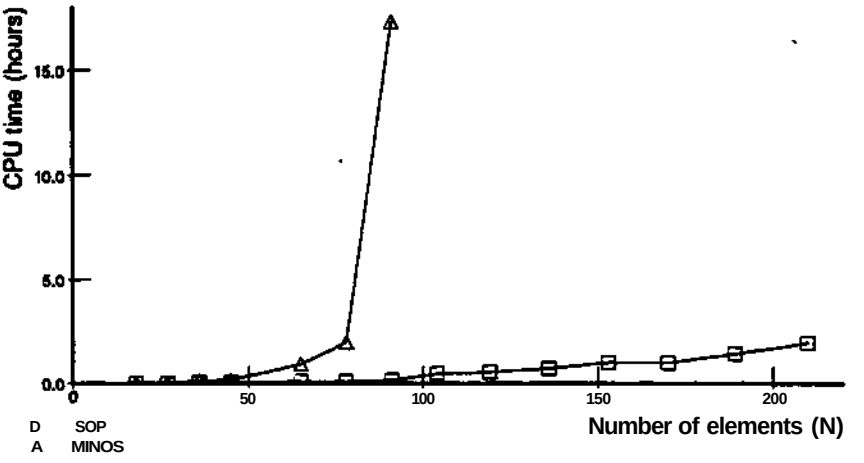

Figure 5.2: MINOS vs. SQP solver, Nested formulation.

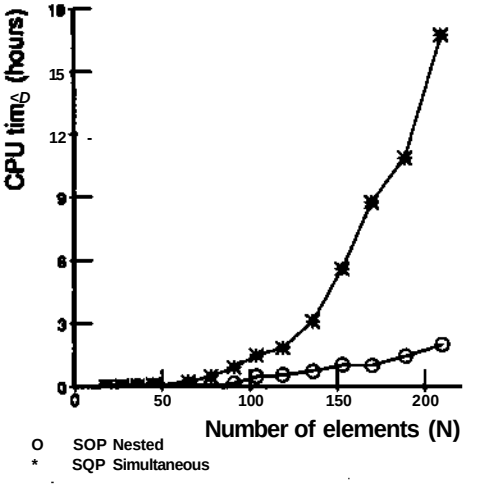

Figure 5.3: Simultaneous vs. nested, SQP solver.

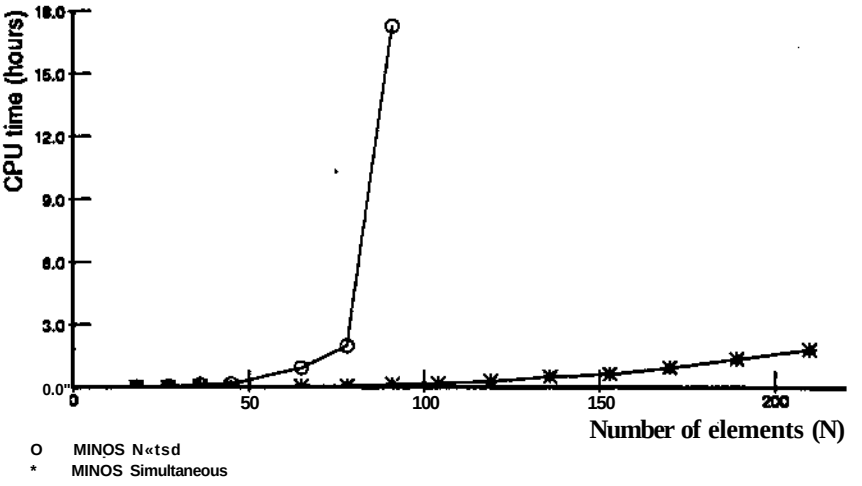

Figure 5.4: Simultaneous vs. nested, MINOS.

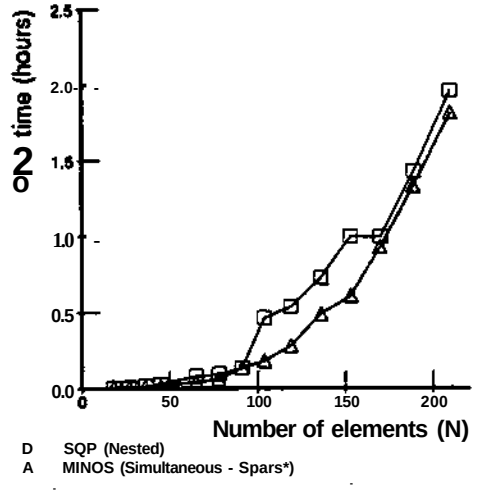

Figure 5.5: Best performing code. (Simultaneous : MINOS ; Nested : SQP)

Chapter 6

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.

Bibliography

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

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

[3] Haftka, R. T., "Integrated Nonlinear Analysis and Design".*AIAA J.*, Vol. 27,1989, pp. 1622-1627

[4] Ryu, Y. S., Haririan, M., Wu, C.C. and Arora, J. S. "Structural Design Sensitivity of Nonlinear Response." *Computers and Structures,* 21, 1985, 245-255.

[5] Wu, C. C. and Arora, J. S., "Simultaneous Analysis and Design of Nonlinear Response", *Engineering with Computers,* Vol. 2, 1987, pp. 53-63.

[6] Wu, C. C. and Arora, J. S., "Design Sensitivity Analysis of Nonlinear Response using Incremental Procedure", *AIAA J.*, Vol. 25, 1987, pp. 1118-1125.

[7] Schittkowski, K. "NLPQL : A FORTRAN Subroutine Solving Constrained Nonlinear Programming Problems.", *Annals of Operations Research,* 5, 1985/6, pp.485-500.

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

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

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

[11] Wakabayashi, Tadaji, "Computational Efficiency and Storage Requirements for the Hierarchical and Simultaneous Methods in Structural Optimization." M.S. Thesis, Department of Civil Engineering, Carnegie-Mellon University, Pittsburgh, PA, 1990.

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

[13] Murtagh, B.A. and Sauders, M.A. "A Projected Lagrangian Algorithm and its Implementation for Sparse Nonlinear Constraints". Report SOL 80-1R, Department of Operations Research, Standford University, California, 1980, in *Math. Prog. Study on Constrained Optimization.*