J. KSIAM **Vol.2**, No.2, 41-47, 1998

SUBSTRUCTURING ALGORITHM FOR STRUCTURAL OPTIMIZATION USING THE FORCE METHOD

HO-JONG JANG
Department of Mathematics
Hanyang University, Seoul 133-791, Korea
hjang@fermat.hanyang.ac.kr

Abstract

We consider some numerical solution methods for equality-constrained quadratic problems in the context of structural analysis. Sparse orthogonal schemes for linear least squares problem are adapted to handle the solution step of the force method. We also examine these schemes with substructuring concepts.

1. Introduction

The problem of computing a sparse self-stress matrix from the equilibrium matrix of a structure arises in the force method of structural analysis. In this context, the system force vector f is a minimizer of the quadratic form corresponding to the potential energy $\frac{1}{2}f^TAf$, subject to the equilibrium equation $Ef = p$. In these equations, E is the underdetermined equilibrium matrix, p is the vector of applied loads, and A is the symmetric, block diagonal, element flexibility matrix.

In the force method, the vector f is computed in two phases:

1. Compute a self-stress matrix N from the equilibrium matrix E, and a particular solution f_p of the equilibrium equation.
2. Solve the system
$$N^TANf_0 = -N^TAf_p, \tag{1}$$
and compute $f = f_p + Nf_0$.

Since the self-stress matrix N is a basis for the nullspace of the equilibrium matrix, we call it a *null basis*. Methods of finding a sparse or structural basis of the nullspace of the equilibrium matrix have been the subject of extensive study over the past few years(see, for example [9]).

In our approach to compute the nullspace we use a parallel scheme by utilizing the graph theoretic ideas in what we call the *substructuring method*, which is introduced

Key Words: Structural optimization, force method, substructuring method, null basis.
AMS subject classification: 65F05, 65F50.
This work was partially supported by Hanyang University Research Grants in 1998.

by Plemmons and White[8]. This method successfully handles parallel nullspace computation. Once the substructures have been identified, then null basis is even readily computable by observation.

In fact nullspace computation of the force method has an important role in the entire force method computation. Let k be the number of columns in N, then the system flexibility matrix N^TAN is a square matrix of order k. In general, the matrix N^TAN is unlikely to be sparse even when N is sparse, and hence it is explicitly computed and stored only for small k. However, by using the special structure inherent in self-stress matrices of the physical structures based on the substructuring method, we can compute the system (1) more efficiently.

The focus of this paper is the use of the substructuring method for phase 2 of the force method while the system (1) is carried out using orthogonal factorization techniques developed for linear least squares problems.

2. Substructuring method

In general, for an equilibrium matrix with m rows and n columns, there exists a product of elementary matrices, P, such that

$$PE = [E_1, E_2] = E_1[I_m, E_1^{-1}E_2],$$

where E_1 is nonsingular. Consequently, the nullspace of PE, and hence E, is generated by the columns of the block matrix

$$N = \begin{bmatrix} E_1^{-1}E_2 \\ -I_{n-m} \end{bmatrix}.$$

With special interest in parallel computation, motivation for developing the substructuring method was that the nullspace computation(forming N) can often be done by appropriate ordering of the nodes and elements, extending certain results in [2]. This ordering yields a matrix E with a great deal of structure which can be exploited by multiprocessing computers in forming N. We briefly consider two classes of structural problems in order to see how the substructuring method works.

EXAMPLE 1. Substructuring with proper partition of a pin-jointed-truss leads to matrix E, which is 20×62, resembling that depicted in Figure 1.

FIG. 1. *Pin-jointed-truss with 6 disjoint substructures and its equilibrium matrix*

The blocks in E are usually sparse(see Plemmons and White [8] for details). Note that each upper triangular diagonal block in E_1 has an inverse, and corresponds to stable substructures given by $\mathcal{S}_1 = (\{1\}, \{e_1\}), \cdots, \mathcal{S}_3 = (\{3, 4, 5\}, \{e_3, e_4, e_5\}), \cdots, \mathcal{S}_6 = (\{10\}, \{e_{10}\})$. The remaining elements, $e_{11}, \cdots, e_{31}$, connect these stable substructures. Note that the diagonal structure of E_2 which is a result of ordering the connecting elements from the left to the right. Because of this structure of E_2, we can also reduce a lot of work in parallel computation of $E_1^{-1}E_2$.

EXAMPLE 2. Consider a rigid frame which models a wheel with 8 spokes in Figure 2. Each spoke is a stable substructure and together they form a proper partition. The matrix E is 96×120.

FIG. 2. *Rigid frame which models a wheel and its equilibrium matrix*

More discussion of the nullspace computation with various examples is in [8].

3. The force method: phase 2

Given a paticular solution f_p of the equilibrium equation, the main task of phase 2 of the force method is the computation of the redundant force vector f_0 which satisfies system (1). System (1) is simply the normal equation for the weighted least squares problem:

$$\text{minimize}_{f_0} \|G^{-1}(Nf_0 + f_p)\|_2, \tag{2}$$

where G is Cholesky factor of element stiffness matrix A^{-1}. Several methods for solving problems of the form (2) are described in [1]. The traditional method of normal equations consists of the direct application of Cholesky's method to the symmetric positive definite matrix N^TAN. Unfortunately, explicitly forming the matrix N^TAN can lead to loss of special block structure of N and worsening of the conditioning of the problem. A better approach in this regard is to apply orthogonal transformations to the matrix $G^{-1}N$, leading to an algorithm of the following form:

$$P_1G^{-1}NP_2^T = Q\begin{bmatrix} R \\ 0 \end{bmatrix}, \; -Q^TP_1G^{-1}f_p = \begin{bmatrix} c \\ d \end{bmatrix}, \; f_0 = P_2^TR^{-1}c, \tag{3}$$

where R is an upper triangular matrix of order $n - m$, P_1 and P_2 are permutation matrices of order n and $n - m$, respectively, Q is an orthogonal matrix of order n, and c and d are vectors of length $n - m$ and m, respectively.

We need to consider the shape of $G^{-1}N$ rather than N^TAN based on substructuring method. The calculation of the QR factorization of $G^{-1}N$ can be done in parallel. The shape of $G^{-1}N$ in Figure 3 is the same block structure as N because G has block diagonal structure.

FIG.3. *$G^{-1}N$ for Figure 2.*

The Givens transformations can be used in parallel by concurrently working on the eight column blocks. The block in row block 33 and column block 1 can be used to annilate the components in row blocks 1-4 and column block 1. At the same time the terms in row blocks 5-8 and column 2 can be annilated by the block in row block 34 and column block 2. The remainder of the top 32 row blocks can be annilated concurrently in a similar manner.

Although orthogonal factorization is numerically superior to the normal equations, poor results may be obtained with either method when the element flexibility matrix A is ill-conditioned. In a series of papers [6, 7], Paige has developed schemes which can considerably reduce this difficulty. Paige's new schemes for linearly constrained sum-of-squares($LCSS$) problem to (2) with trunback-QR factorization for nullspace computation are discussed in [3]. Furthermore, these new schemes successfully applied to rapid reanalysis of structures, and reported in [4]. In our approach, each of the two formulations of Paige's method is reviewed by using the substructuring method with special structure of the nullspace N.

Formulation I. Following Paige [6], if we define the weighted residual vector

$$v = G^{-1}(Nf_0 + f_p),$$

then problem (2) can be written in the equivalent form

$$\mathrm{Min}_{v,f_0} v^T v \quad \text{subject to} \quad Gv = Nf_0 + f_p. \tag{4}$$

First, decompose N in (4) as

$$Q^TN = \begin{bmatrix} Q_1^TN \\ Q_2^TN \end{bmatrix} = \begin{bmatrix} R \\ 0 \end{bmatrix}, \tag{5}$$

where R is a nonsingular upper triangular matrix of order $n-m$, $Q=(Q_1,Q_2)$ is an orthogonal matrix of order n, and Q_1 and Q_2 are $n\times(n-m)$ and $n\times m$ matrices, respectively. The constraints in (4) then split into

$$Q_1^T Gv = Rf_0 + Q_1^T f_p \tag{6}$$

$$Q_2^T Gv = Q_2^T f_p. \tag{7}$$

Since R has full row rank, (6) can always be solved for f_0 once v is given, and so (7) gives the constraints on v, and (4) becomes

$$\text{Min}_v v^T v \quad \text{subject to} \quad Q_2^T Gv = Q_2^T f_p.$$

Next, apply the QR factorization to $(Q_2^T G)^T$ starting from the lower right components to decompose $Q_2^T G$ so that

$$Q_2^T GP = (0, L_2),$$

where $P=(P_1,P_2)$ is an orthogonal matrix of order n, and P_1 and P_2 are $n\times(n-m)$ and $n\times m$ matrices, respectively. Here L_2 has full column rank.
That is, decompose $Q^T G$ as

$$Q^T GP = \begin{bmatrix} Q_1^T GP_1 & Q_1^T GP_2 \\ 0 & L_2 \end{bmatrix} = \begin{bmatrix} L_1 & L_{12} \\ 0 & L_2 \end{bmatrix}, \tag{8}$$

Assuming L_2 is nonsingular we now obtain

$$v = P_2 L_2^{-1} Q_2^T f_p,$$

since $Q_2^T GP_2 = L_2$. Finally, f_0 is recovered from the triangular system (6).

The advantage of using substructuring in the formulation I occurs in (5). Since N is more sparse than $G^{-1}N$, we need to have relatively less computation compare to (3). However, the first formulation does not take advantage of any special structure the matrix G may have, G will be triangular if it is computed by Cholesky factorization, and in our case G has block diagonal structure. Indeed, that structure is in general destroyed by the computation (6)-(7). This problem can be resolved in Paige's alternate formulation of *LCSS*.

Formulation II. Paige[7] has given a formulation II in which the two orthogonal transformations U and V (to the corresponding matrices Q and P in (8), respectively) simultaneously in a manner which retains the triangular structure of G throughout the computations. For implementation details, see [7].

The result is a factorization of the form

$$U^T[f_p, N, GV] = \begin{bmatrix} f_{p1} & 0 & M_1 & 0 \\ f_{p2} & S^T & M_{21} & M_2 \end{bmatrix}, \tag{9}$$

where S^T, M_1, and M_2 are lower triangular matrices of order $n-m$, m, and $n-m$, respectively, and $U=(U_1,U_2)$ and $V=(V_1,V_2)$ are orthogonal matrices of order n

and m, respectively. The matrices U_1 and V_1 are $n \times m$, and the matrices U_2 and V_2 are $n \times (n-m)$. We note that the matrices in (9) are not necessarily identical to the corresponding matrices in (8). Applying transformation (9) and using the change of variable

$$V^T v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix},$$

problem (4) becomes

$$\mathrm{Min}_{v_1,v_2} v_1^T v_1 + v_2^T v_2 \quad \text{subject to} \quad \begin{bmatrix} M_1 & 0 \\ M_{21} & M_2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 \\ S^T \end{bmatrix} f_0 + \begin{bmatrix} f_{p1} \\ f_{p2} \end{bmatrix}.$$

Thus v_1 is completely determined by the equation

$$M_1 v_1 = f_{p1},$$

and the functional is minimized by taking $v_2 = 0$. Finally, the solution f_0 may now be determined from the system

$$S^T f_0 = M_{21} v_1 - f_{p2}.$$

The reduction in (9) can be carried out in two main stages. The first stage consists of $n-m$ major steps in each which one computes the superdiagonal of the (f_p, N) matrix is annihilated while the lower triangular form of the G matrix is preserved, in our case G is already triangular and block diagonal. For example with m=1, n=4, the second step is as in Figure 4.

FIG. 4. *Second step of the zero chasing for (9).*

The rotations are ordered 1,1$'$,2,2$'$, and the nonzero element i^*, introduced by rotation i from the left, is immediately made to be zero by rotation i' from the right. When the first stage has been completed, each step of the second stage has the same form, eliminating one more complete diagonal of the (f_p, N) matrix, and maintaining the lower triangular form of the N matrix. Thus in the second stage there will be $m-1$ major steps. Since the shape of N is block banded and there are not many nonzero components of the superdiagonal of the N, we do not need much computation in the first stage. For the second stage computation, the Givens transformations can be used in parallel by concurrently working on each column block. Hence, the lower triangular S^T from N based on the substructuring method can be computed in parallel without much of work.

4. Concluding remarks

In this paper we have attempted to describe the effects of using the substructuring and the force method for solving the constrained minimization problem arising from structural optimization. The special structure of the self-stress matrix N enables us to save a great deal of work in the computations, and also allows us to use parallel computational techniques.

The algorithm we have described are currently being implemented on current multiprocessing computer. The results of these numerical tests and comparisons will be reported in detail elsewhere.

In fact we also investigated the substructuring method for the solution step of the force method in the context of the incompressible fluid flow. A number of interesting characterizations of this problem and some parallel iterative schemes are reported in [5].

References

[1] A. Bjorck, *Numerical Methods for Least Squares Problems*, SIAM, 1996.

[2] J.R. Gilbert and M.T. Heath, *Computing a sparse basis for the null space*, SIAM J. Alg. Disc. Meth., Vol.8, No.3 (1987), pp. 446–459.

[3] M. Heath, R. Plemmons and R. Ward, *Sparse orthogonal schemes for structural optimization using the force method*, SIAM J. Sci. Statist. Comp., Vol.5, No.3 (1984), pp. 514–532.

[4] H. Jang, *A constrained least squares approach to the rapid reanalysis of structures*, Linear Algebra Appl., Vol.265 (1997), pp. 185–202.

[5] H. Jang and R. White, *Sparse matrix method for equilibrium equation*, in Proceedings of the Copper Mountain Conference on Iterative Methods, T. Manteuffel and S. McCormick, eds., Copper Mountain, Colorado, 1998.

[6] C.C. Paige, *Computer solution and perturbation analysis of generalized linear least squares problems*, Math. Comput., Vol.33, No.145 (1979), pp. 171–183.

[7] C.C. Paige, *Fast numerically stable computations for generalized linear least squares problems*, SIAM J. Numer. Anal., Vol.16, No.1 (1979), pp. 165–171.

[8] R. Plemmons and R. White, *Substructuring methods for computing the nullspace of equilibrium matrices*, SIAM J. on Matrix Anal. and Applic., Vol.11, No.1 (1990), pp. 1–22.

[9] J.M. Stern and S.A. Vavasis, *Nested dissection for sparse nullspace bases*, SIAM J. on Matrix Anal. and Applic., Vol. 14, No.3 (1993), pp. 766–775.