Finite Element Analysis in Concurrent Processing: Computational Issues

Jaroslaw Sobieszczanski-Sobieski[*]
Langley Research Center, Hampton, VA 23681

Brian Watson[†] and Garrett Vanderplaats[‡]
VR&D, Inc., Colorado Springs, CO 80906

The purpose of this research is to investigate the potential application of new methods for solving large-scale static structural problems on concurrent computers. It is well known that traditional single-processor computational speed will be limited by inherent physical limits. The only path to achieve higher computational speeds lies through concurrent processing. Traditional factorization solution methods for sparse matrices are ill suited for concurrent processing because the null entries get filled, leading to high communication and memory requirements. The research reported herein investigates alternatives to factorization that promise a greater potential to achieve high concurrent computing efficiency. Two methods, and their variants, based on direct energy minimization are studied: a) minimization of the strain energy using the displacement method formulation; b) constrained minimization of the complementary strain energy using the force method formulation. Initial results indicated that in the context of the direct energy minimization the displacement formulation experienced convergence and accuracy difficulties while the force formulation showed promising potential.

I. Introduction

THIS report summarizes the results of a research to investigate the potential of alternate methods for solving large-scale structural problems on future computers. The finite element method has proven to be a widely used technique to perform structural analysis. The method allows continuum problems of solid mechanics to be approximated by a finite number of unknowns. Using variational principles of mechanics, static problems reduce to the solution of a system of linear equations (or a sequence of linear equations for non-linear problems). The need for detailed analyses of complex structural systems (aerospace, automotive, and civil engineering) will soon lead to structural models with 10 to 100 million degrees of freedom.

Solution of such large problems in a reasonable time will require computing at the processing rate far exceeding the capacity of today's single processor CPU (Central Processing Unit)-based computers. CPU speeds will eventually reach inherent physical limits as processor miniaturization descends to the molecular and atomic scales. Recent computer technology advances (large-scale multi-processors, clusters or grids of commodity CPU's, and reconfigurable computers such as Field Programmable Gate Arrays) are being developed to improve aggregate computing speeds. Although widely different in their details, these technologies share the common theme of *concurrent processing*.

However, engineers cannot use the new computing technology until they have applications software developed that exploits the technology to its full potential. Traditional methods (algorithms) cannot realize the full potential of concurrent processing because they are rooted in the sequential thinking of the human brain. The method commonly used in the Finite Element Analysis (FEA) solves the load-deflection equations, [K] {U} = {P}, for displacements U

[*] Sr. Research Scientist, Structures & Materials, NASA Langley Research Center, MS240, Hampton, VA 23681, AIAA Fellow.

[†] President, Vanderplaats Research and Development, Inc., 1767 S. 8th St, Colorado Springs, CO 80906, AIAA Fellow.

[‡] Project Manager, Vanderplaats Research and Development, Inc., 1767 S. 8th St, Colorado Springs, CO 80906, AIAA Sr. Member.

by use of one of the standard algorithms, e.g., pivoting or triangular factorization, applied to the sparse matrix of coefficients (the stiffness matrix).

This method lacks efficiency in concurrent processing because in the factoring operation a significant percentage (20% is typical according to[1]) of the null entries in the sparse K become non-zero (a filling-in phenomenon) in a pattern difficult to predict in advance. When the factoring operation is partitioned among many processors, the filling-in complicates memory allocation to the individual processors. Furthermore, the factoring partitions remain coupled, resulting in the time-consuming, inter-processor communication.

The alternative of the domain decomposition (e.g., analysis by substructuring) creates, on one hand, the opportunity to engage many processors, but, on the other hand, it requires additional computing effort to account for substructure couplings. That additional effort detracts from the parallelism gain increasingly as the number of substructures increases. Paradoxically, in the limit when each substructure shrinks to just one finite element all the gain is erased. Hence, the experience indicates that the law of diminishing returns limits the number of concurrently operating processors to low hundreds[2]. In contrast to the matrix factoring, the iterative methods[3,4] based on reduction of the load-deflection equations residual are free of the filling-in. In the concurrent processing, they have a potential, in principle, to engage as many processors as there are individual terms in the equations. However, they require a problem-dependent number of iterations.

This study focuses on the class of iterative methods that, instead of reducing the load-deflection equations residual, minimize directly the structure elastic strain energy. The methods investigated herein fell into the following categories: a) Unconstrained minimization of energy as a function of the displacement variables; b) Minimization of energy as a function of the elemental forces under equilibrium constraints; c) Minimization of energy as a function of the elemental forces, unconstrained by prior elimination of the equilibrium equations.

The candidate methods evaluated using example applications on a single processor computer with the immediate purpose to determine which of the candidate algorithms perform well enough in terms of accuracy and convergence characteristics to justify future testing on a multiprocessor computer. The testing eliminated the first two algorithm categories. Their discussion and results are provided in the Appendix for the sake of completeness. Preliminary results for the third category were encouraging enough to qualify it for further research and for the detailed description in the body of the report.

II. Direct Energy Minimization Formulations

To avoid the drawbacks of the traditional formulation, the guiding idea behind this research was:

"Avoid forming and factoring large matrices."

Without pivoting or factoring, there is no fill-in to increase storage and complicated inter-processor communications. Thus, storage requirements will increase linearly with the number of nodes.

This idea leads back to the fundamental variational principles of mechanics. These principles allow a structural problem to be cast as an energy minimization problem. The principle of minimum of energy is well established in literature[5,6] and direct minimization has been proposed before, but it has not been practical because there was no optimization method effective for the very large number of state variables encountered in structural analysis. That obstacle has been recently removed and optimization with hundreds of thousands of variables is now possible[7]. This enabled the formulations adopted herein. The Direct Energy Minimization (DEM) replaces solution of the load-deflection equations, [K] {U} = P by one of the alternatives presented below.

The Direct Energy Minimization in a Displacement Method (DEM/DM) formulation is written as

$$\text{GIVEN:}\quad \text{structure geometry, material properties, and loading forces } P\hat{}\qquad (1a)$$

$$\text{FIND:}\quad \{U\}\qquad (1b)$$

$$\text{MINIMIZE:}\quad 1/2\ \Sigma_k \left(u_k^{e\ T} K_k^e u_k^e\right) - P^{\wedge T}\, U \qquad (1c)$$

where the elemental displacements $\{u^e\}$ relate to the nodal displacements {U} by the connectivity matrix A which contains only one unity entry in each row

$$u = A\ U \qquad (1d)$$

The counterpart of Eq. (1) stated in the form of the Direct Energy Minimization in the Force Method formulation, DEM/FM may be cast as:

GIVEN: structure geometry, material properties, and loading forces P (2)

FIND: f, the vector of all elemental forces (2a)

MINIMIZE: $1/2\ \Sigma k \left(f_k^{e\ T} F_k^e f_k^e\right)$ (2b)

SATISFY: $AT\ f = P$ (2c)

$Hk\ f_k^e = 0$ for all elements (2d)

where A, again, is the connectivity matrix; f_k^e - forces on k-th element in the Displacement Method format; P - loads that include reactions (self-equilibrated loads); and $H_k\ f_k^e = 0$ represents equilibrium equations for k-th element.

The DEM/FM is being considered for the following reason. The elemental stresses are of primary interest in many applications and they derive directly from the elemental forces f^e. In the DM formulation these forces are secondary variables obtained by differencing the displacement field. Inherently, then, the displacements must be calculated to accuracy higher than the accuracy acceptable for stress. A simple rod under axial load, shown with all the data in Fig. 1, illustrates this point. The rod is divided into N finite elements, numbered starting with 1 at the fixed end. The displacement at the loaded end is u_N = PL/EA = 4.8**5**4133294500270. For N = 1000, the displacement at the junction of the elements N and N-1 is u_{N-1} = 4.8**4**9279161205770. The difference between u_N and u_{N-1} begins at the third significant digit. Repetition of the calculation for N = 10,000, 100,000, and 1000,000 shows that the first difference location migrates to the 4-th, 5-th, and 6-th significant digit. In the DEM context, the above observation translates into expectation of the FM formulation converging to the accuracy satisfactory for stress in the number of iterations smaller than in the DM formulation. This may offer a significant advantage considering that stress seldom needs to be defined to more than three to four significant digits for meaningful comparison with the material allowables typically available with three significant digits precision.

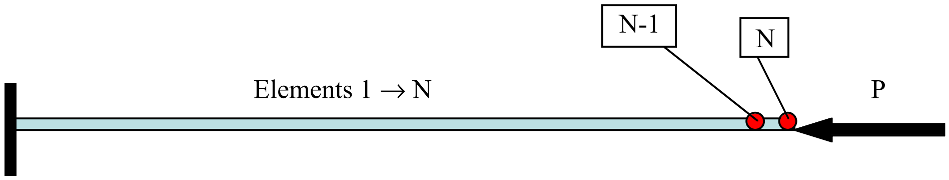

Figure 1: A simple rod under axial load modeled with N; Length L =10000 cm; P = 10000 Newton; E = 20601000 N/cm sq.

As with DEM/DM, DEM/FM also affords an opportunity for concurrent computing in the element-by-element evaluation of the elastic strain energy, Eq. (2b).

In this study, both DEM/DM and DEM/FM formulations were examined by numerical testing. As it is well-known, the equality constraints are usually an obstacle to fast convergence especially in large problems. Therefore, the presence of the equality constraints in DEM/FM was addressed in three variants of the solution of Eq. (2):

a) Application of SUMT (Sequential Unconstrained Minimization Technique);
b) Partial elimination of the equality constraints; and
c) Complete elimination of the equality constraints.

The results for DEM/DM and the variants a and b of DEM/FM were found to be inadequate to continue. For the sake of the report completeness, these formulations and their results are included in the Appendix.

The DEM/FM variant c formulation generated results encouraging enough to warrant further attention and development. Its detailed description follows.

III. DEM/FM with Complete Elimination of Equality Constraints

Description of the DEM/FM variant c begins with Eqs. (2c) and (2d). It entails the equilibrium considerations, transformation of A^T, and reduction to unconstrained minimization problem.

A. Equilibrium analysis

The load vector {P} in Eq. (2c) is a superposition of the applied load {P^} and a set of statically determinate reactions R_s (redundant reactions will be introduced later) induced by P^. The reactions, R_s, are obtained from the body equilibrium equation

$$[E_B]\ \{R_s\} = \{P\hat{}\} \qquad (3)$$

where matrix E_B is, typically, 3×3 for 2D structure cases, and 6×6 in 3D cases. When R_s becomes known, P is computed from

$$P = P\hat{} + R_s \qquad (4)$$

Another way to represent P is

$$P = p\ \{P^n\} \qquad (5)$$

where p – a parameter, and P^n – normalized vector.

Table 1 defines dimensions for the matrices and vectors in Eqs. (2)-(5). The difference between NU and NNE stems from the assumption that P is a self-equilibrated load vector that includes reactions. Construction of such a vector implies solution of the body equilibrium equations that are equivalent to a subset of the equilibrium equations for the joints (nodes) by the theorem of equivalence between the equilibrium of the parts and the equilibrium of their assembly.

Table 1. Dimensional data definitions

	Number of :
EDOF	Elastic degrees of freedom (total)
NFT	Elemental (internal) forces (total)
NNE	Nodal equilibrium equations
NB	Body equilibrium equations
NFU	Elemental forces reduced by the use of elemental equilibrium equations
NU	Usable nodal equilibrium equations = NNE – NB
NEE	Single element equilibrium equations = EDOF
NFE	single element forces
NFI	Single element independent forces = NFE – NEE
NFD	Single element dependent forces = NEE

In a *statically determined* structure, NFU = NNE – NB so the forces f may be obtained by solution of Eqs. (2c) and (2d) without the aid of the energy minimization in Eq. (2b). In a *statically undetermined* structure where NFU > NU, a constrained minimization, i.e., entire Eq. (2), is required. In principle, one may eliminate the equality constraints by merging Eqs. (2c) and (2d) into a single set of equations. Pivoting on that set[8] leads either to a complete solution for f if the structure is statically determinate (then no further processing by DEM is necessary), or to partitioning of f into f^d-dependent forces, and f^i-independent forces, if the structure is redundant. However, this approach requires forming and factoring large, sparse matrices, which should be avoided according to the guiding principle of this study.

The following operations replace the above matrix factorization. Having the elemental forces in Eq. (2d) partitioned into f^d, and f^1, subsets for each element

$$f_k^e = \{f^d | f^i\}_k \qquad (6)$$

Eq. (2d) is solved for f^d to obtain

$$f_k^d = E_k\ f_k^i \qquad (7)$$

As the above relationship holds for each element, E is saved as a part of the element definition (matrices E for elements of the same type are all the same, except of the embedded geometrical dimensions that may differ from one element to the next).

B. Transformation of A^T

Turning attention to Eq. (2c), A^T contains unity entries representing f^i and f^d. Its standard form in a graphic rendition is depicted in Fig. 2. Now, use of Eq. (7), eliminates from A^T those columns that correspond to f^d. By permutation of the remaining columns, one may convert Eq. (2c) to a form illustrated in Fig. 3 where LT is a lower triangular submatrix associated with the dependent forces f^d, C is a rectangular submatrix associated with the independent forces f^1, and B is a submatrix pertaining to the rigid body equilibrium.

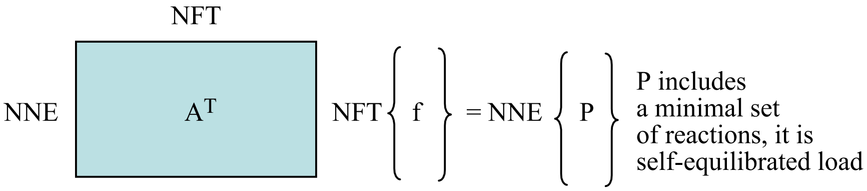

Figure 2: Assembled structure equilibrium equation.

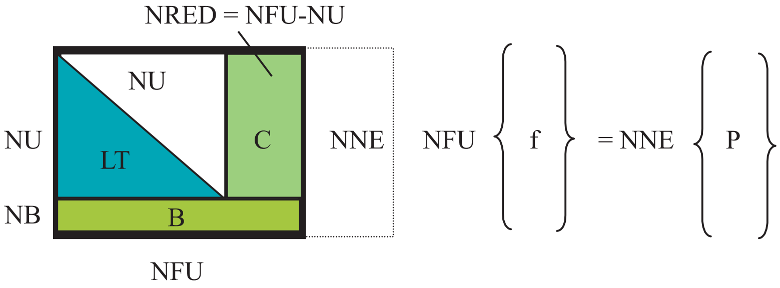

Figure 3: Assembled structure equilibrium equation transformed.

Before describing the above transformation details, it should be clarified that the presence of [LT] in Fig. 3 brings about one key advantage. *Using [LT], it is now possible to solve Eq. (2c) by a single forward substitution (FS) without use of any matrix factoring and without filling-in that would reduce the matrix sparsity.*

As an analogy one may refer to Cholesky's algorithm to solve a generic set of simultaneous, linear equations [M] {x} = {b} by factoring [M] into the lower-triangular and upper-triangular matrices, LT and UT. Under that algorithm {x} is then obtained by executing one Forward Substitution (FS) on LT and one Backward Substitution (BS) on UT. Similarly, the transformation from the format in Fig. 2 to Fig. 3 generates LT but without the matrix factoring and without creating UT, so that BS is unnecessary.

It was found that the LT pattern in A^T is not obtainable in the exceptional case when a finite element exists in which NFI < NFD, (Table 1), as it occurs in truss rods and shear panels. The simple remedy is, for example, to treat the rod as a beam, and to achieve a pure truss action, if necessary, by prescribing very small moments of inertia.

1. Details of the A^T Transformation

Consider a generic A^T and examine a submatrix representing contributions of the k-th element. Fig. 4 shows A^T with the k-th element submatrix highlighted (heavy line box) for an arbitrary case for which there are NFE = 8 element forces numbered on the top. These forces contribute to the EDOF equilibrium equations corresponding to the matrix rows (highlighted green) as indicated by unity entries. By definition there is only a single entry of 1 per column, all other entries are zeros. The pattern of the non-zero locations results from the numbering of the Finite Element Model nodes and the EDOF within an element and may not exhibit any particular regularity. In the ensuing transformations, the row numbering does not change but the columns are permuted, first within each element submatrix and, next, over the entire matrix.

The number of dependent forces at the element level is denoted NEE. Let NEE = 3 in the k-th element, hence NFI = 8 - 3 = 5. The dependent forces are placed at the end (red) of the numbered list. Figure 5 illustrates how by

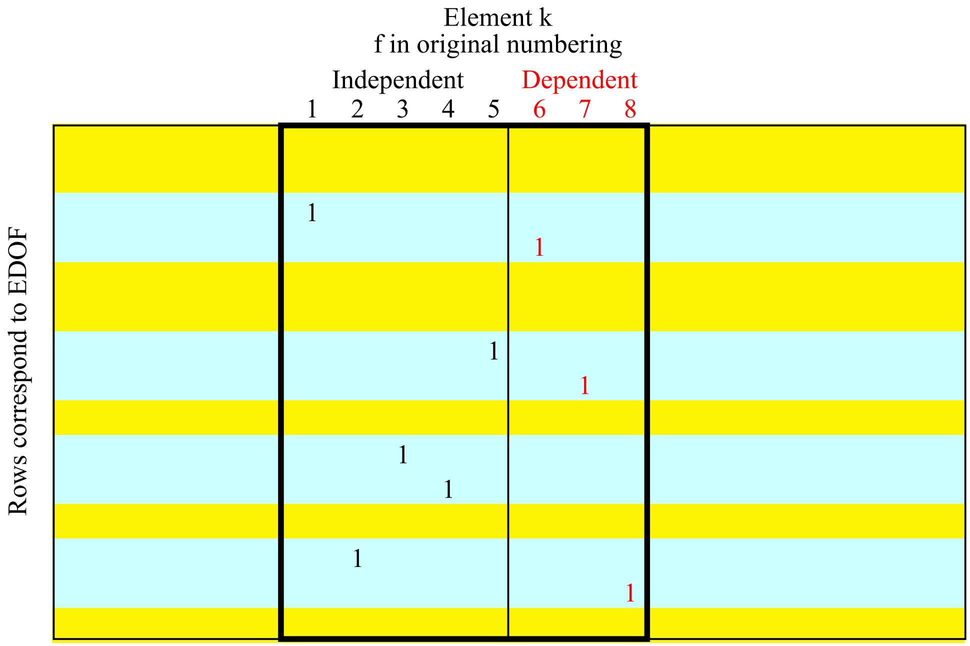

Figure 4: A^T matrix with submatrix corresponding to k-th element (heavy line box).

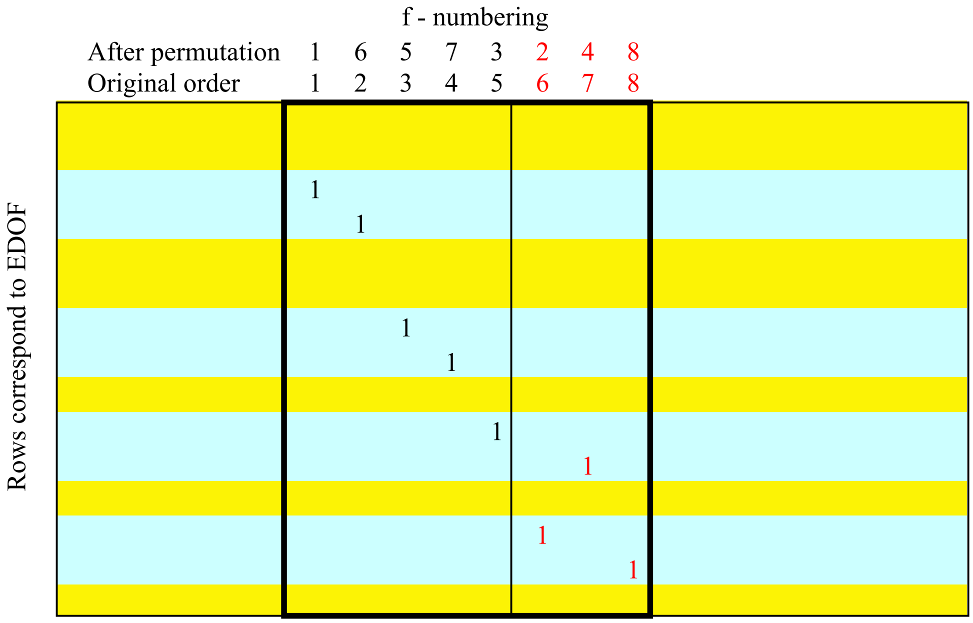
f - numbering
After permutation 1 6 5 7 3 2 4 8
Original order 1 2 3 4 5 6 7 8
Rows correspond to EDOF
1
1
1
1
1
1
1
1

Figure 5: Submatrix for k-th element permutated to for a NW-SE sequence of 1's.

permuting the columns within the element submatrix, the unity entries may be relocated to form a sequence of 5 starting at the NW corner and extending toward SE in a regular pattern such that each next entry is below the previous one. The remaining 3 entries (highlighted red in the figure) do not follow in the above pattern.

The sequence of NFI = 5 identifies the element f^i, and the remaining NFD = 3 entries constitute the element f^d in a new order. It will become apparent soon that the location of the f^d entries *below* the f^i entries is important. The two lists of numbers on top show how the forces migrated from the original to new positions. This completes the force renumbering at the element level.

Next, recall that f^d depends on f^i per Eq. (7). Therefore, it is possible to delete the column corresponding to f^d and to replace each f_i^d unity entry with the influence coefficient representing $f_i^d(f_j^i)$ placed in the same row and the column f_j^i. The solutions of Eq. (7) for f^d obtained for the unity entries on the right hand side, taken one at a time, supplies the influence coefficients to be referred to as the f^d coefficients.

Figure 6 depicts the k-th element submatrix after the column deletion. It shows the NW-SE sequence of the unity entries and the blocks highlighted by a dot-pattern that hold the f^d coefficients.

The above rearrangement repeated for all elements (it may be all done simultaneously) completes the first step of the transformation. It results in A^T whose number of columns is smaller, and its nonzero entries are distributed as depicted in Fig. 7. The sequences of 1's are in the grey blocks, the f^d coefficients are contained in the dot-pattern blocks located *below* the sequences of 1's within each submatrix. While the column permutations in the foregoing

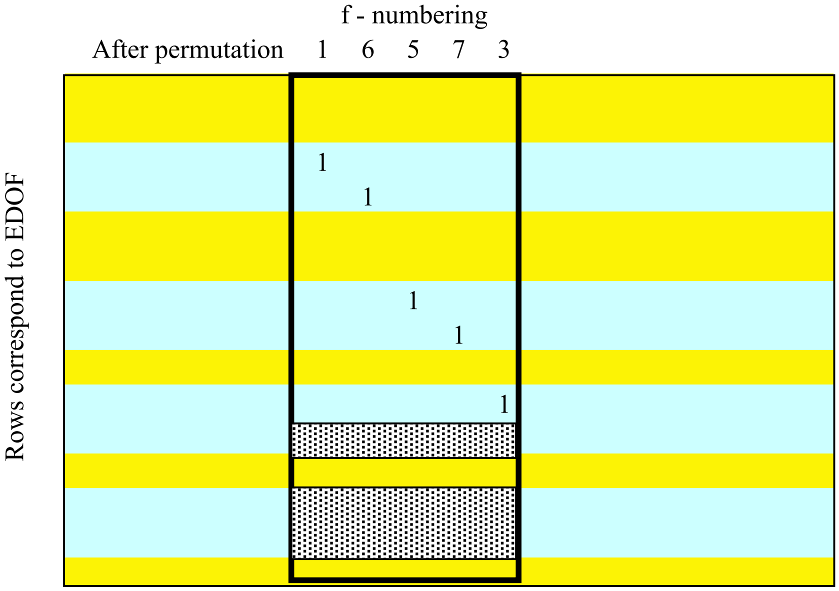

Figure 6: Dependent force columns deleted in the submatrix. Pattern-filled blocks hold dependent force coefficients.

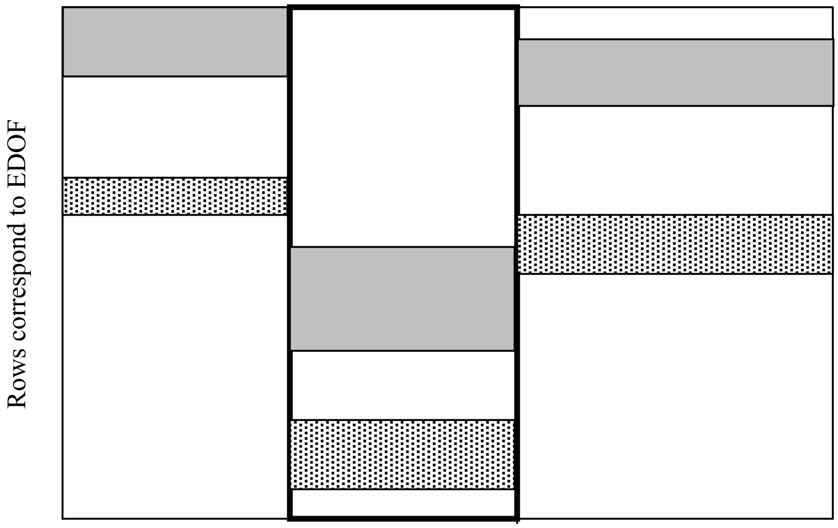

Figure 7: Global view of entire A^T (only three element submatrices in this example) after column permutation and deletion within each submatrix. Pattern-filled blocks hold the dependent force coefficients.

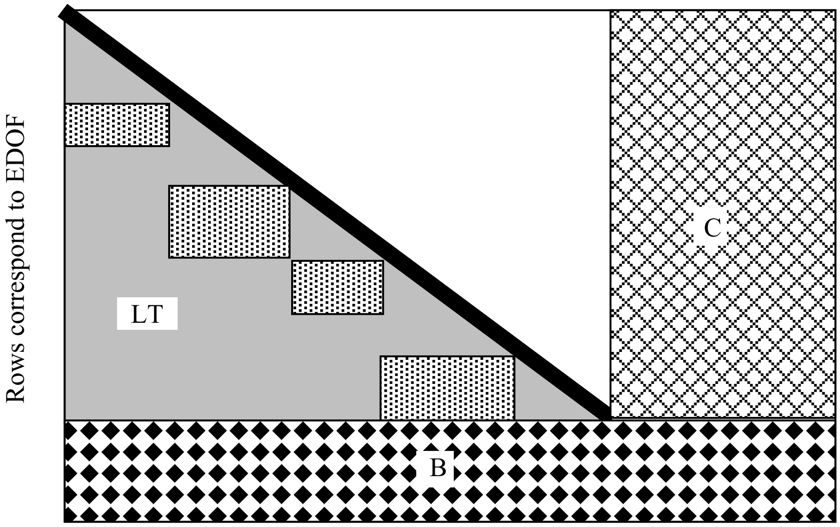

Figure 8: Matrix A^T after column permutation to construct the NW-SE line (bold) of 1's and submatrices LT, C, and B. Dot pattern-filled blocks hold dependent force coefficients.

occurred within each element submatrix, now, in the second step of the transformation, the columns are permuted matrix-wide, disregarding the element submatrix boundaries, to construct a continuous sequence of unity entries extending from the NW corner SE-ward. As each column contains a single unity entry and a group of the f^d coefficients *below* it, all these coefficients end up *below* the diagonal sequence of unity entries. Figure 8 illustrates the new pattern of the nonzero entries.

The formation of the diagonal sequence of unity entries stops when no more unity entries can be found available to be permuted. The diagonal sequence of 1's (the bold line in Fig. 8) together with the f^d coefficients beneath it (dot-pattern blocks) form the [LT] submatrix. The rows remaining below [LT] constitute the submatrix [B], and the submatrix remaining in the NE area is [C] (which vanishes in a statically determinate structure). The new order of columns defines a new numbering of f^e whose correspondence to the original one is saved to allow the original form of F^e in the energy minimization in Eq. (2b).

In actual programming, the entire matrix transformation operation is implemented by manipulating the indices of sparse arrays.

This transformation identifies automatically f^d and f^i at the assembled structure level (while Eq. (7) does that at the element level). At the assembled structure level, f^d relates now to f^i and P* by

$$[LT]\ f^d = -[C]\ f^i + \{P^*\}; \tag{8}$$

where * signifies that P is curtailed by clipping the bottom NB entries. The dimensions in Eq. (8) are: [LT], NU×NU; $\{f^d\}$, NU×1, [C], NU×(NFU-NU); $\{f^i\}$, (NFU-NU)×1; and {P*}, NU×1. For a statically determined structure, C vanishes and f^d may be obtained from the remainder of Eq. (8).

In summary, the elimination of the dependent forces in the above transformation is a two-steps process. First, the dependent forces are eliminated at the element level using Eq. (7). Next, permuting the columns and reducing A^T in Fig. 2 to LT, C, and B in Fig. 3 finds more dependent forces. In general, in each of these two steps, the numbering of the elemental forces changes and the changes must be recorded to enable return to the original numbering for the final result output.

2. Dependent Forces as Functions of the Independent Ones

Because the ultimate goal is the unconstrained energy minimization, Eq. (2b) must be stated in terms of f^i as independent variables. It is useful, therefore, to obtain an explicit relationship of f^d to f^i and P

$$f^d = [D_f]\ f^i + f_P^d \tag{9}$$

where f_P^d is f^d due to P.

To obtain $[D_f]$ and f_P^d one solves Eq. (8) by superposition, separately for unit values of f^i, and separately for P*, remembering that DEM must be repeated for all the loading cases. The step-by-step algorithm is:

Algorithm: step-by-step: (10)

a) Set P*= 0, and solve [LT] f^d = - [C] f^i for one of the elements in $\{f^i\}$ set to 1, all other elements set to 0;
b) Record the solution $\{f^d\}$ as a column in $[D_f]$;
c) Repeat from step #a for another element in $\{f^i\}$, until $[D_f]$ is filled;
d) Save $[D_f]$ for the use in the energy minimization;
e) Set f^i = 0, and solve [LT] f^d = P*; and record the solution as f_P^d.
f) Repeat step e) for NLC loading cases.

The above algorithm provides an opportunity for using concurrent processing in step #c whose repetitions may be executed all at once. Furthermore, the entire algorithm needs to be carried out only once before the minimization begins.

3. Redundant Reactions

Until this point the presence of any redundant support reactions was ignored. However, if they exist they must be treated in the energy minimization, Eq. (2b), as additional independent variables alongside f^i. Formalization of this may be accomplished by defining {R}, NR×1, as the vector of all supports reactions. Then, R is partitioned into

$$\{R\} = \{R_s | R_r\} \tag{11}$$

where R_s is the set of reactions minimally needed to eliminate the body freedom (for example, 3 reactions for a 2D case, and 6 reactions for a 3D case), and R_r, NRR×1, contains the excess redundant reactions.

Analogous to Eq. (7), the body equilibrium equation for a particular redundant reaction R_{rk}

$$[E_B]\{R_s\} = \{\hat{R}_{rk}\} \tag{12}$$

yields R_s induced by $\hat{R}_k$. A superposition of $\hat{R}_{rk}$ and R_s it induces forms a vector $\{R_s\}$ for the above particular R_{rk}. This vector may also be represented in a form similar to Eq. (5):

$$R_r = p_r\{R_r^n\} \tag{13}$$

where p_r is a parameter, and the R_r^n vector is normalized.

Treating R_r as loads acting on the structure in addition to P, Eq. (9) generalizes to

$$[LT]\{f^d\} = -[C]\{f^i\} + p_r\{R_r^n *\} + \{P*\}; \tag{14}$$

By setting f^i = 0, and P* = 0, and repeating the solution of Eq. (14) with p_r = 1 for all redundant reactions R_r one at a time, one generates columns of matrix D_r needed to augment Eq. (9) with a term $[D_r]p_r$ representing R_r

$$f^d = [D_f]f^i + [D_r]\{p_r\} + f_P^d \tag{15}$$

where $\{p_r\}$ contains parameters of all the redundant reactions (Eq. (13)).

On a historical note, in the Force Method practiced before the digital computers became common the statically indeterminate unknowns such as f^i and R_r were exposed by fictitiously "cutting" the structure. The placement of cuts was judgmental. In the Finite Element version of the Force Method, the cutting was automated, and the judgment was removed, by pivoting on the equilibrium matrix to identify the redundant unknowns[8,9,10]. In that context, the foregoing introduces yet another way for identification of the redundant unknowns. The common feature of all these techniques is non-uniqueness of the identification of the redundant unknowns.

4. Generating Elemental Flexibility Matrices

Generation of the elemental flexibility matrices from the stiffness matrices available in a standard finite element code may be accomplished in two ways. One way is to perform a partial factorization with full pivoting of the element stiffness matrix. The factorization stops when all remaining diagonals are zero (or numerically negligible). This process outputs the elemental flexibility matrix defined relative to the remaining pivots which are designated as supported[10]. No judgmental preselection of the supported degrees of freedom is needed. The other way is computationally simpler but requires a preselection of a set of supports sufficient to remove the element rigid body degrees of freedom. Then, the element stiffness matrix is modified by zeroing out the rows and columns corresponding to the supported EDOF and placing unity entries at the diagonal where these rows and columns intersect. This modification[8] removes the element stiffness matrix singularity so the matrix may now be inverted. The formation of the element flexibility matrix is completed by substituting zeros at the diagonal locations in the inverted matrix where the unity entries were previously placed in the stiffness matrix.

5. Unconstrained Minimization

With the relationship of Eq. (15) available, the energy minimization, Eq. (2), takes on the following unconstrained form:

FIND: $f^{\,i}$ and p_r, i.e., the independent elemental forces and redundant reaction parameters; (16)

MINIMIZE: $1/2\ \Sigma_k\left(f_k^{e\ T}\ F_k^e\ f_k^e\right)$ (16a)

USE: $f^e=\{f^d|f^i\}$, where $f^d=f^d\left(f^i,\ p_r\right)$ per Eq. (15). (16b)

Solution of Eq. (16) yields $f^{\,i}$ and p_r whose substitution into Eq. (15) generates $f^{\,d}$. The reactions are recovered by computing R_r from Eq. (13), and obtaining R_s by superposition of two terms: R_s due to P^ from Eq. (3), and R_s generated by all individual R_r per Eqs. (13) and (12).

6. Computing the Displacements

Unlike the Displacement Method, the Force Method does not yield the entire displacement field. The displacements of interest must be selected and, then, calculated by means of the dummy unit load (DUL) in a virtual work method. This method produces a particular displacement U_i as:

$$U_i=\Sigma_k\left(f^e_{DUL\ k}{}^{T}\ F_k^e\ f_k^e\right) \tag{17}$$

where f^e_{DUL} stands for the forces generated on k-th finite element by DUL associated with the particular U_i.

In Eq. (17), f_k^e is available from solution of Eq. (16). Given a DUL, the forces f^e_{DUL} may be obtained from solution of equilibrium of the structure converted to a statically determinate one; satisfaction of the compatibility equations is not necessary. This means that one may construct the equilibrium equation patterned after Eq. (14)

$$[LT]\ \{f^e_{DUL}\}=d\{S^n*\} \tag{18}$$

where the right hand side has a format similar to the one in Eq. (5), i.e., the unit load is represented in the same manner as load P by S^n standing for a normalized vector that contains d = 1 for the DUL itself and R_s that DUL induces, and by setting d = 1 by the definition of DUL. The * at S^n has the same meaning as the * at P in Eq. (8).

If MU displacements are desired for NLC loading cases, the sequence of Eqs. (17) and (18) must be repeated MU×NLC times. Again, multiprocessing computing may be used to compress the elapsed time for this sequence as well as for the element-by-element operation in Eq. (17).

IV. Multi-span Beam Test Case

A 6-span beam loaded by concentrated moments as shown in Fig. 9 provides an example of application and a test for Eq. (16). Table 2 displays the bending moment values at the supports obtained by DEM/FM and the reference values obtained from the standard 3-moments equations (that derive from the Castigliano Theorem). The problem's geometrical symmetry and its loads anti-symmetry were deliberately not taken advantage of in the DEM/FM solution to gain an additional assessment of the accuracy by comparing forces that ought to be equal due

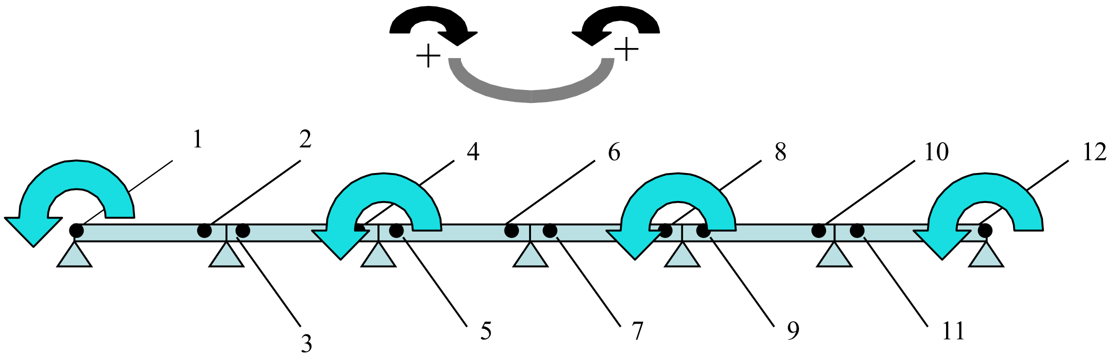

Figure 9: A 6-span beam. Inset shows positive bending moment. Table 4 displays bending moment values at the numbered points.

Table 2. Results for 6-span beam test case

Point no. (see Fig. 9)	Bending moment DEM/FM (a)	Reference exact format (b)	Reference decimal format	Relative error |(a-b)/a|
1	−1	−1	−1	0
2	0.13330997	2/15	0.13333333	0.00018
3	−0.13331	−2/15	−0.1333333	0.00017
4	0.46666867	7/15	0.46666667	4.3E-06
5	−0.53333133	−8/15	−0.53333333	3.8E-6
6	1.9984E-05	0	0	0.00015*
7	−1.998E-05	0	0	0.00015*
8	0.53333523	8/15	0.53333333	3.6E-06
9	−0.46666477	−7/15	−0.46666667	4.1E-06
10	0.1333566	2/15	0.1333333	0.00017
11	−0.13335665	−2/15	−0.13333333	0.00017
12	1	1	1	0

* Normalized by max(Moment) = 8/15

to the symmetry. The Table 2 results show that the relative error with respect to the reference solution was less than 0.02% and the relative error in symmetry was less than 0.007%. The process converged in 5 iterations using the optimizer available in the EXCEL spreadsheet.

This test case was too small to extrapolate its results to large-scale applications. Also, it did not provide empirical data on concurrent computing advantages because it was implemented on a conventional, single processor computer. However, by converging to accurate results, the test indicated that DEM/FM has a potential to warrant its further investigation.

V. Conclusions

The study examined a finite element solution method which uses direct energy minimization as an alternative to the conventional load-deflection equation formulation. The motive is to exploit power of concurrent computing in the large-scale structural analysis. The experience and theory reported in literature point to the scalability limits of the conventional methods based on the sparse matrix factorization. In contrast, the direct energy minimization (DEM) method potential to fit the concurrent processing technology stems from its inherent parallelism predicated on its use of element-by-element processing which avoids forming and processing large sets of simultaneous equations.

- Five alternative formulations for the DEM method were investigated by numerical testing. The following four exhibited performance judged to be inadequate;
- Displacement formulation;

- Force formulation with equilibrium equality constraints;
- Force formulation with partial elimination of the equality constraints;
- Force formulation with the equality constraints incorporated through a penalty function.

The poor performance of the above formulations was traced to ill conditioning, lack of a universally effective and application-independent technique for preconditioning in the conjugate gradient optimization, and inability to satisfy the equilibrium equality constraints with acceptable accuracy.

In contrast, the fifth alternative, the DEM in a Force Method formulation, with the equality constraints completely eliminated turned out to be promising. The removal of the equality constraints was achieved by solving the equilibrium constraints using the matrix column permutation to reduce the matrix to a lower-triangular form without factoring. This approach overcame the difficulties experienced with the first four formulations and has generated results that were good enough to warrant further investigation for very large-scale applications on a multiprocessor computer. The latter will provide the actual measure of improvement because the metrics of elapsed time, memory requirements, and cost strongly depend on the details of the particular hardware architecture used.

VI. Appendix—DEM/DM and DEM/FM: Formulations and Testing Results

This Appendix contains the formulation details and results for the methods categorized in the Introduction as: a) Unconstrained minimization of energy as a function of the displacement variables; b) Minimization of energy as a function of the elemental forces under equilibrium constraints.

A. DEM/DM Formulation

The problem is set as an unconstrained minimization defined in Eq. (1). The optimization was conducted by means of the conjugate gradient method aided by preconditioning[10]. An example of a solution successfully obtained by DEM is a solid cube whose FEM is depicted in Fig. A1. This 3-D brick model simultaneously illustrates the best performance of the conjugate gradient method and the worst performance of direct factorization. The model consists of 8000 elements. Using commercial finite element and optimization codes described in[11,12], this model was solved with the conjugate gradient method in only 37 sec on a single processor computer. This compares very favorably to the 161 sec required by a direct factorization on the same single processor computer. However, for more realistic models, particularly those with shell elements, the conjugate gradient method was found to not perform well. A remedy could, potentially, be provided by a suitable preconditioner. Unfortunately, as discussed in Ref. 10, there is no universally effective technique for constructing a preconditioner so that a case-by-case experimentation is involved. Table 3 shows comparison of the performance of the conjugate gradient method without any preconditioning for the cube model (#1) in Fig. A1, as well as several other shell and solid models originating from the recent industrial practice. Their size in terms of EEDOF ranges from the order of thousands for Model #2 to hundreds of thousands for Model #6.

The elapsed time for DEM in a Conjugate Gradient search is not fully indicative of the method performance potential because a single processor computing was used for all results in the table. The comparison on strain energy shows agreement ranging from exact, #1-3, to very poor as in #6. The residual taken as a synthetic measure of satisfaction of the load-deflection equations is 3 to 6 orders of magnitude worse than in the conventional method.

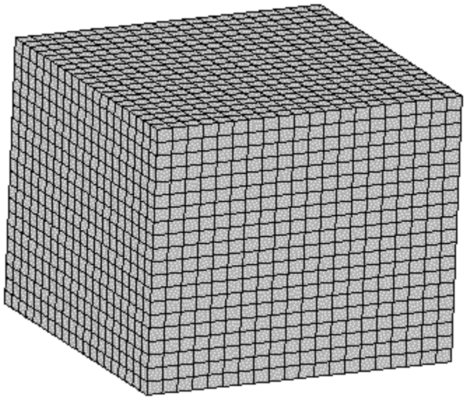

Figure A1: Solid cube modeled with brick 3-D elements.

Table 3. Sample single-processor conjugate gradient timings

Problem name	Number of DOF	Conjugate gradient				Factorization		
		Iterations	Residual	Strain Energy	Time (sec)	Residual	Strain Energy	Time (sec)
1. cube	27771	143	3.367379E-10	1.651754E+00	37.00	6.916198E-14	1.651754E+00	161.00
2. cclip	3681	351	−1.737790E-12	9.307911E+01	4.00	−2.057317E-14	9.307914E+01	3.00
3. tete	20082	1465	−2.484554E-11	1.104309E+06	92.00	−7.405986E-14	1.104309E+06	21.00
4. knuckle	14544	2680	−1.420550E-07	1.881993E+03	118.00	−4.189363E-12	1.882032E+03	20.00
5. kenki	59643	11929	−2.620755E-07	5.695816E+05	1776.00	3.291284E-13	5.917549E+05	259.00
6. hardtop	248970	49795	3.168668E-07	4.047854E+01	37724.0	3.934094E-11	7.284592E+01	472.00

Considering the previous discussion of the accuracy of the elemental forces relative to the displacements in DM, the comparison on the residuals in Table 3 does not qualify the DEM in the DM formulation (DEM/DM) without a preconditioning as a tool for use with confidence. It would be premature to conclude that an upgrade of the DEM/DM algorithm to the accuracy adequate for practical purposes for both displacements and forces is not possible. However, such upgrade would require development of a new preconditioner universally effective across the range of commonly encountered types of structures. Such development being beyond the scope of this study, the DEM/DM formulation was set aside, and attention in the remainder of the study turned to the DEM in a Force Formulation (DEM/FM) discussed in detail in the body of the paper.

B. DEM/FM Formulation

A formal statement for DEM/FM is given in Eq. (2). By its nature, this is an equality-constrained minimization and different ways of accounting for the constraints engender different techniques for solution. Two solutions tried herein are: representing the constraints as an objective function penalty, and a partial elimination of the equality constraints in two versions. Version 1 calls for solving the elemental equilibrium Eq. (2d), and Version 2 partially eliminates the equality constraints by solving the assembled structure equilibrium Eq. (2c).

1. Penalty Function Formulation

This formulation uses the Sequential Unconstrained Minimization (SUMT) algorithm. A quadratic penalized objective function was constructed to which an efficient quadratic minimizer, e.g., a conjugate gradient technique, could be directly applied. The penalized objective function, Φ, was written as

$$\Phi = 1/2\ \ f^{e\,T} F^{e}\, f^{e} + \mu\,(A^{T} f^{e} - P)^{T} [\lambda]\,(A^{T} f^{e} - P) \qquad \text{(A1)}$$

where μ is the overall penalty parameter, and [λ] is a diagonal matrix of penalty coefficients λ_I assigned to individual constraints.

Results of attempting to apply this method to a 10-rod truss (defined in Ref. 8, p. 197) showed great sensitivity to the penalty parameters. The extreme ill conditioning of the resulting system caused the conjugate gradient method to have difficulty keeping new search directions conjugate to the previous directions. Conclusion reached from the above results was similar to that drawn for DEM/FM: to make this approach useful would require new development of a preconditioning technique for the conjugate gradient method. Consequently, the SUMT approach to DEM/FM was not pursued any further.

2. Elimination of Elemental Equilibrium Equations

Version 1 of this approach calls for using Eqs. (6) and (7) to eliminate f^{d} and rewriting Eqs. (2b) and (2c) in terms of the remaining f^{i}; Eq. (2d) drops out. Version 2 accomplishes the similar result by permuting the columns of A^T matrix in Eq. (2c) to obtain

$$[I|A^{TP}]\ \{f^{e}\} = \{P\}; \qquad \text{(A2)}$$

where f^{e} is partitioned into f^{i} and f^{d}.

The permutation of A^T to $[I|A^{TP}]$ is possible because, by definition, a particular f^{e} contributes to only one of the nodal equilibrium equation. Hence, each column of A^T is all null except a single entry of unity.

Solving Eq. (A2) for f^d leaves Eq. (2) curtailed to Eqs. (2a), (2b), and (2c). In general, each of these two versions may result in a different allocation of the elements of f^e to f^d and f^i and different length of the f^d and f^i vectors. Neither approach eliminates the equality constraints completely, so both are a compromise between the ideal of a completely unconstrained optimization and the need to avoid factorization of large matrices.

Both of the above versions have been implemented using commercial grade codes. The GENESIS[11] finite element code generated the element stiffness matrices, and the BIGDOT[12] solved the resulting constrained minimization problem. BIGDOT allows constrained minimization, but all of the constraints must be of the inequality type. To enforce the equality constraints into a form acceptable to BIGDOT, each equality was expressed as a pair of opposite-in-sign inequalities.

3. Rod 1-D Test Case

All the numerical experiments herein were performed on a single processor computer because the immediate purpose was to determine which of the candidate algorithms perform well enough in terms of accuracy and convergence characteristics to justify future testing on a multiprocessor computer. Initial experiments showed that the optimizer had a difficult time in enforcing all of the constraints. Additionally, since the optimizer enforced constraints only to within a set tolerance, small errors could propagate and aggregate across the model. A one-dimensional rod model with 5000 elements, previously introduced in Fig. 1, illustrates the case. The structure being statically determinate, there was a unique set of the feasible forces, and the only task for the optimizer was to satisfy the constraints.

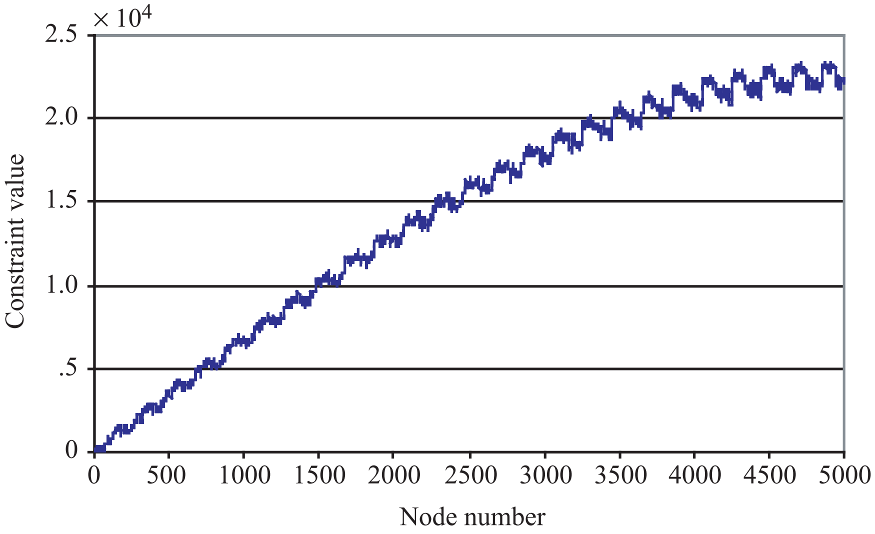

Figure A2: Nodal force imbalance constrain.

Figure A2 shows the nodal imbalances for the final forces obtained by the optimizer. It can be seen that these imbalances are well below the constraint tolerance. Figure A3 shows the element force error as a function of the element number. At the point of application of the load, the element force is extremely accurate. However, the nodal imbalances accumulate to create errors that soon reach unacceptable levels.

As one strategy to reduce the error accumulation from slight nodal imbalances, the reaction force was included along with the applied load. Because this system is statically determinate, the reaction force could be obtained from inspection. To apply this to more general models, additional constraints are imposed to require that the reaction forces be in equilibrium with the applied loads. Figures A4 and A5 show that even with an increased constraint tolerance, this simple change leads to a dramatic reduction in the error propagation, though it is still unacceptably high.

For the second strategy to reduce the element force errors, a residual correction step was proposed. For any linear system:

$$Ax = b \tag{A3}$$

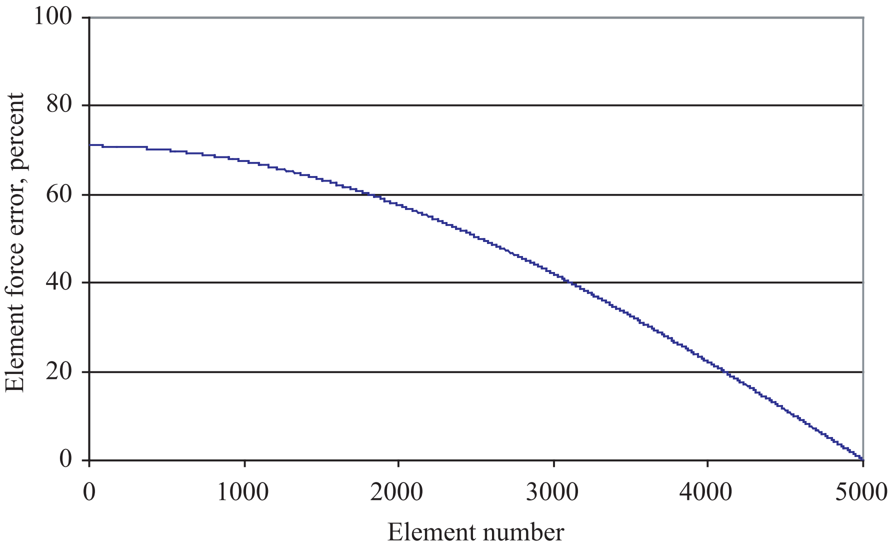

Figure A3: Element force error distribution over the rod length.

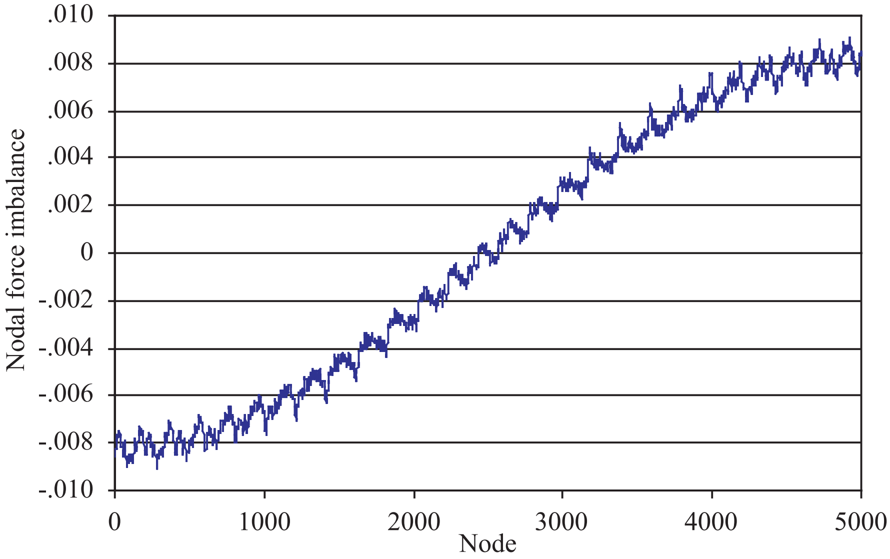

Figure A4: Nodal force imbalance.

for which there is a capability to approximate the solution by $\bar{x}$:

$$\bar{x} = \bar{A}^{-1}b \tag{A4}$$

where $\bar{A}^{-1}$is an approximation to A^{-1}.

A residual-based iterative correction procedure can be constructed as:

$$\begin{aligned} r &= b - Ax_i \\ x_{i+1} &= x_i + \bar{A}^{-1}r \end{aligned} \tag{A5}$$

This procedure converges if and only if the spectral radius of $(I - \bar{A}^{-1}A)$ is less than unity (thus an approximation to A^{-1} may only be used if it is “close enough”).

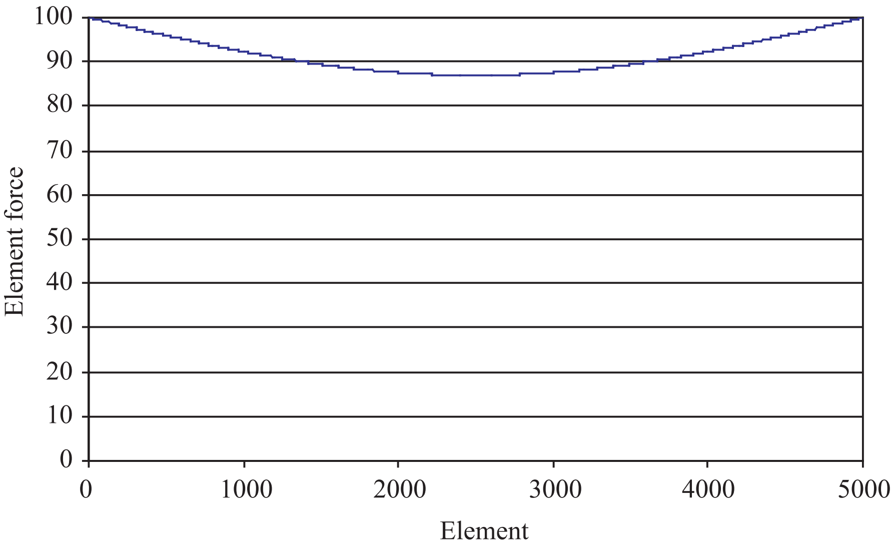

Figure A5: Element axial force distribution.

Because the optimizer is approximately satisfying a system of equality constraints, it can be viewed as an operator that approximates the inverse of the coefficient matrix. Thus, the above residual correction procedure can be applied. After the initial optimizer solution, the nodal force imbalances are gathered and treated as a new corrective loading. The element forces obtained from the corrective loading are added to the original element forces to obtain new corrected element forces. This process could continue iteratively until a desired accuracy is reached. Numerical experiments with the 1-D rod example showed that one residual correction step was sufficient to bring all of the element forces to less than 0.01% error.

4. Plate 2×3 Test Case

Version 1 of the DEM/FM solution including both the reaction forces and the residual correction step was applied to a simple cantilever plate model consisting of 6 QUAD4 shell elements (Fig. A6). Numerical experiments have shown that proper scaling of both the element force variables and the constraint values is crucial to obtaining accurate results. In particular, because the rotational forces (moments) tend to have a different order of magnitude that the translational forces, they must be scaled differently. Otherwise, the optimizer tends to favor one set of forces over the other, and does not converge to accurate results. Figure A7 shows that with proper scaling, accurate results can be obtained for this small model. In fact, the residual correction step was not necessary in this case.

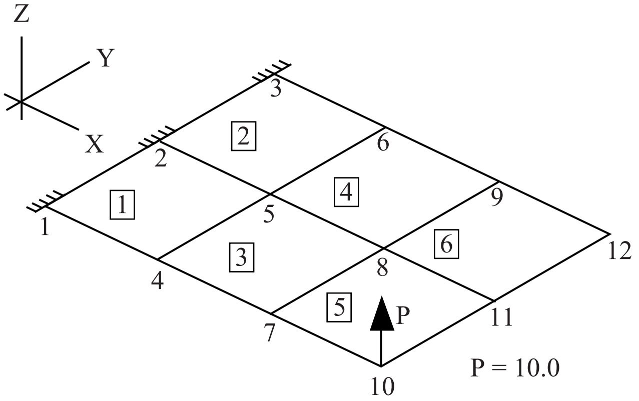

Figure A6: 2×3 plate.

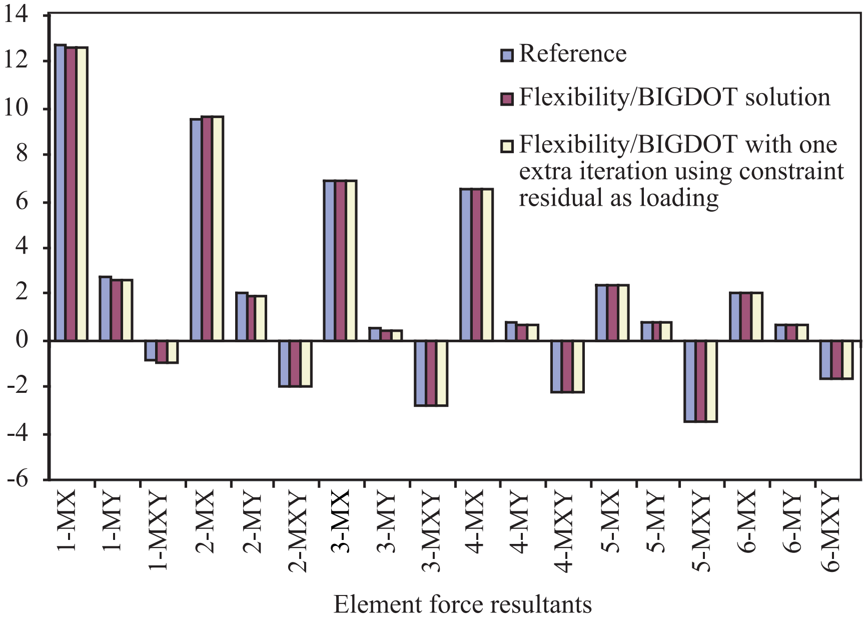

Figure A7: 2×3 plate resultant forces.

5. Box Beam Test Case

Following the good results from the simple cantilever plate model, a somewhat more complicated shell model was constructed. This model of a tapered box beam with a shear web contains more coupling of bending and in-plane forces (see Fig. A8). The results show that even with the improvements of including the reaction forces, the residual correction step, and proper scaling, accurate results were not obtained (see Table 4). The optimizer was able to satisfy the constraint equations, but then failed to reduce the objective function. It is likely that proper scaling the objective function value with respect to the constraint values could remedy the difficulty.

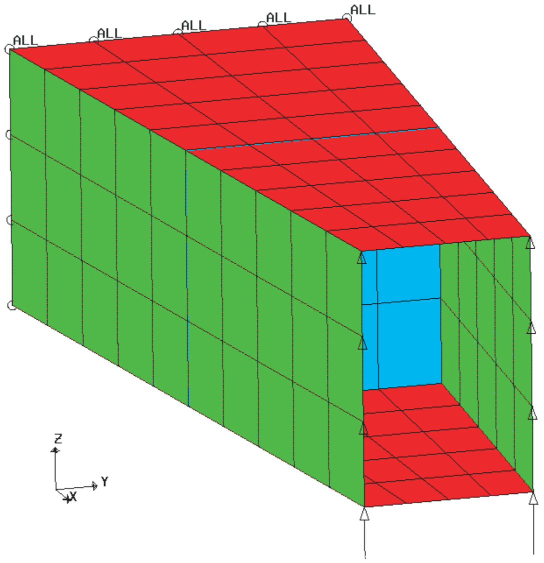

Figure A8: A thin-walled box beam.

Table 4. Sample of element force resultants for cantilevel box beam (Fig. A8)

Current BIGDOT/FM Results				Reference Results			
QUAD4 ID	-NX- -MX-	-NY- -MY-	-NXY- -MXY-	QUAD4 ID	-NX- -MX-	-NY- -MY-	-NXY- -MXY-
1	3535.791 192.5922	−3159.08 693.1301	−258.509 908.5403	1	11.66792 0.014871	63.89381 0.079455	4.80409 −0.00218
2	1185.675 −93.3975	−855.517 3210.543	−82.5887 977.2568	2	−1.27423 −0.0118	58.70311 0.018378	−3.71174 0.015368
3	−119.044 79.02282	69.79244 4335.385	7.82654 555.2748	3	−0.7418 −0.005504	53.40004 0.02482	−5.98264 0.011424
4	−474.539 −149.442	211.8338 3814.969	31.015 203.3365	4	−2.57632 −0.007741	48.90023 0.01677	−6.49746 0.006012
5	−494.157 −138.465	164.6404 4182.982	31.3083 32.17092	5	0.440966 −0.0037	43.8262 0.015109	−8.08197 −0.00224

6. *Plate 10×10 Test Case*

To test version 2 for dependent force variable elimination, a 10×10 mesh of quad4 elements with simple supports on the four edges was constructed (Fig. A9). This model was subjected to uniform normal pressure. This method showed results of similar quality to version 1. Figure A10 shows that while the trend of the MX and MY force resultants are captured, the magnitude is off by about 10%. The MXY force resultants show significant error.

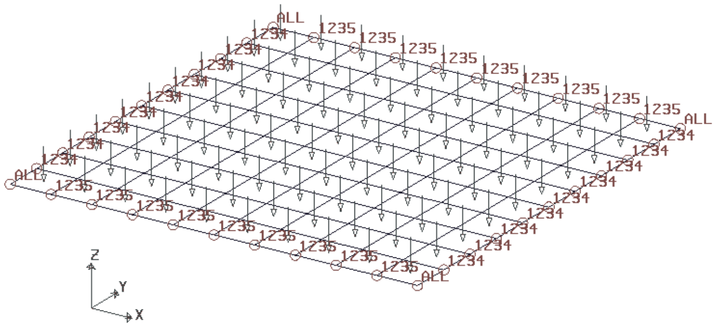

Figure A9: 10×10 plate.

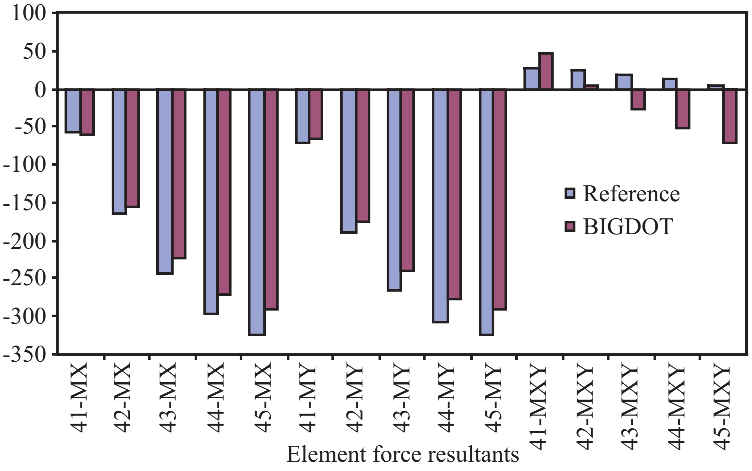

Figure A10: 10×10 plate resultant forces.

VII. References

[1] Nguyen, Duc T., “Parallel-Vector Equation Solvers for Finite Element Engineering Applications”; Kluwer Academic Publishers, 2001.

[2] Storaasli, O., Nguyen, D., and Agarwal, T., “Parallel-Vector Solution of Large-Scale Structural Analysis Problems on Supercomputers”, *AIAA J.*, Vol. 28, No. 7, July 1990, pp. 1211-1216.

[3] Poole, G., Liu, Y. C., and Mandel, J., “Advancing Analysis in ANSYS through Solver Technology”, Electronic Transactions on Numerical Analysis, Vol. 15, 2003, pp. 106-121.

[4] Van der Vorst, H., *Lecture Notes on Iterative Methods*, Report TR/PA/92/75, CERFACS, Toulouse, 1992.

[5] Gallagher, R., *Finite Element Analysis Fundamentals*, Prentice-Hall, 1975.

[6] Zienkiewicz, O., and Taylor, R., *The Finite Element Method*, McGraw-Hill, Vol. 1, 1989, and Vol. 2, 1991.

[7] Vanderplaats, G., *Very Large Scale Optimization*, NASA SBIR Contract Report NAS1-0012, May 2002.

[8] Przemieniecki, J. S., “Theory of Matrix Structural Analysis”, McGraw-Hill Book Co., 1968.

[9] Robinson, J., “Integrated Theory of Finite Element Methods”, J. Wiley & Sons, 1973.

[10] Hughes, T., *The Finite Element Method – Linear Static and Dynamic FE Analysis*, Prentice-Hall, 1987, pp. 483-489.

[11] Anon., *GENESIS Structural Analysis and Optimization Software User Manual,* Version 7.2, VR&D, 2002.

[12] Anon., *BIGDOT Very Large Scale Design Optimization Tool Users Manual,* Version 1.0*,* VR&D, 2002.