ELSEVIER

Comput. Methods Appl. Mech. Engrg. 149 (1997) 319–337

Computer methods in applied mechanics and engineering

# A Direct Flexibility Method

C.A. Felippa*, K.C. Park

*Department of Aerospace Engineering Sciences and Center for Aerospace Structures, University of Colorado, Campus Box 429, Boulder, CO 80309-0429, USA*

Dedicated to Professor J. Tinsley Oden on occassion of his 60th birthday

**Abstract**

We present a Direct Flexibility Method (DFM) for the solution of finite element equations. This method is based on a decomposition of the finite element model into substructures, which may reduce to individual elements. Substructures are preprocessed by the Direct Stiffness Method (DSM) to generate free–free flexibility matrices for floating substructures. The interface problem is solved for the interface forces and the solution recovered over substructure interiors. The DFM shares with the DSM the advantages of being automatic, maintaining locality and sparseness, efficiently handling continuum elements, and requiring only the availability of element stiffness libraries. The new method appears to be advantageous for specific applications. These include: massively parallel processing, inverse problems, treatment of rigid members and inclusions, and use of underintegrated elements without spurious-mode stabilization.

## 1. Introduction

This exposition consolidates material dispersed in previous reports and papers on flexibility methods in structural mechanics [1–6]. Those developments were motivated by needs of specific applications: inverse problems for damage detection, localized vibration control, and massively parallel computations. The underlying theme was the use of techniques of partitioned analysis originally developed for coupled problems [7–9].

By now there are sufficient common ingredients in that material to piece together the basic steps of a solution method identified as Direct Flexibility Method or DFM. Notice that our title is 'a DFM' rather than 'the DFM'. In fact, this is an instance of a class of methods generated from a general variational flexibility formulation presented in [6] and outlined in Section 8. That general formulation includes the well-known Classical Force Method (CFM) as well as the present DFM as special cases. The departure from the CFM emerges by exploiting two attributes of the Direct Stiffness Method (DSM) version of the Displacement Method: use of free–free element matrices, and element-by-element assembly. The motivation for combining features from both CFM and DSM has rich historical roots. In the following we highlight historical points relevant to the present exposition.

The pre-computer form of the CFM had a long and distinguished history since the source contributions by Maxwell, Mohr and Castigliano. It was a favorite of experienced structural engineers because it provides directly the internal forces, which are of paramount interest in stress driven design. Because of its physical transparency of this form, which relies on the hand selection of redundant forces through appropriate cuts or releases, is still taught in introductory courses in structures.

Semi-automatic matrix forms of the CFM evolved after World War II with the appearance of digital computers [10–14]: the analyst still selects the force redundants but the resulting matrix equations were solved

* Corresponding author.


PII S0045-7825(97)00048-0

by computer. As structural models increased in size and complexity while competition with the DSM heated up, methods for fully-automated, computer-based selection of redundants were developed in the 1960s [15–17]. These developments did not, however, spare the Force Method from extinction when the Finite Element Method (FEM) spread beyond the aerospace industry through general-purpose codes [18]. The CFM had always been at its best for skeletal structural models: trusses and frameworks, in which there is a close relation between internal and nodal forces. It does continuum elements clumsily. A key computational deficiency is that numerically stable computer selection of redundants, as needed to compete against the fully automatic DSM, hinders the sparsity of the solution matrices. This makes a big difference as the size of continuum models grows. For example, in a $10 \times 10 \times 10$ mesh of 8-node bricks the CFN-to-DSM solution time ratio exceeds one million.

Since 1970 several investigators [19–27] have continued research in the Force Method for selected applications such as structural optimization. Those efforts have concentrated on two different areas. Activity continued in the CFM aimed at extracting a sparse null basis of the equilibrium matrix so as to produce a sparse symmetric redundant-flexibility matrix. This line of research, pursued by linear algebrists [19–22], appears to have been closed by 1990. Patnaik and coworkers [23–25] have developed a non-classical approach called the Integrated Force Method, which maintains sparsity for continuum elements at the expense of symmetry.

Mathematically, the fundamental procedure adopted in the CFM is to decompose the solution of the governing equilibrium equations into particular and homogeneous parts. Physically, these decompositions lead to the statically determinate and indeterminate contributions to the complete solution. Regardless of interpretation, not only the member flexibility matrices but also the assembled flexibility matrix pertaining to the statically indeterminate structure must have full rank, a property that hinders locality and sparsity.

The present DFM retains several aspects of the CFM, such as symmetric equations and the use of flexibility matrices. The main deviation is in the problem decomposition methodology. In the present DFM a structure is partitioned into a number of substructures. Substructures may be *floating*, that is, contain rigid body motions. An important consequence of this attribute is that, for each substructure, the corresponding flexibility and stiffness matrices become dual of each other. This endows the present DFM with the locality and sparseness enjoyed by the DSM. To achieve those goals we need to introduce new concepts and tools that are absent in the CFM, in particular rigid body motions, self-equilibrium conditions and the free–free flexibility. Thus, the present DFM shares many of the element-by-element processing features of the DSM, and in fact can make full use of standard finite element stiffness libraries.

Work in the present DFM was initially motivated by applications to inverse problems in structural mechanics as well as massive parallel processing. An attractive, scalable parallel solution approach called FETI (Finite Element Tearing and Interconnecting) has been developed since 1990 by Farhat, Roux and coworkers. Formulation and applications of FETI methods are described in a recent comprehensive survey [28]. In terms of partitioned analysis procedures [7–9], FETI may be interpreted as a *differentially partitioned* solution procedure in which a large finite element model, embodying possibly millions of equations, is decomposed into nonoverlapping subdomains. Subdomains are connected by discrete or distributed Lagrange multipliers, which represent interaction forces. Each subdomain is mapped to a processor. The interior problem of disconnected subdomains is solved by the displacement method via local sparse solvers, whereas the interface connection problem is treated by a preconditioned, projected conjugate gradient solver.

An alternative formulation of the FETI methods, called *algebraically partitioned* FETI or A-FETI [3,4], has been found to have deeper connections with the Force Method. That work, as well as related investigations into system identification and damage localization [1] led to a general variational derivation of a spectrum of flexibility-based methods [6]. The name 'Direct Flexibility Method' emphasizes the use of a new definition of flexibility matrix, which exists for floating elements or substructures. This *free–free flexibility* is dual to the well known *free–free stiffness matrix* that is the building block of the Direct Stiffness Method.

We realize that the use of the qualifier 'direct' in the DFM may be subject to differing interpretations [29]. In the DSM, 'direct' refers to the immediate merge of free–free element stiffness matrices into the master stiffness matrix as elements are being formed. In the present paper 'direct' refers to the availability of free-free substructural flexibilities that are dual to the free-free stiffness matrices. These free-free substructural flexibility matrices need not be assembled when using parallel iterative algorithms such as the A-FETI method, thus their locality is preserved.

## 2. The Classical Matrix Force Method

This section reviews the basic steps of the matrix CFM for treating a linear finite element model. The five governing matrix equations are

$$\begin{aligned}
&\text{Equilibrium:} && \bar{\boldsymbol{p}} = \boldsymbol{B}_0\bar{\boldsymbol{f}} + \boldsymbol{B}_1\bar{\boldsymbol{x}} \\
&\text{Constitutive (flexibility form):} && \bar{\boldsymbol{v}} = \bar{\boldsymbol{F}}\bar{\boldsymbol{p}} \\
&\text{Compatibility:} && \boldsymbol{B}_1^{\mathrm{T}}\bar{\boldsymbol{v}} = \boldsymbol{0} \\
&\text{Displacement-deformation:} && \bar{\boldsymbol{u}} = (\boldsymbol{B}_0 + \boldsymbol{B}_1\boldsymbol{X})^{\mathrm{T}}\bar{\boldsymbol{v}} \\
&\text{Redundant forces:} && \bar{\boldsymbol{x}} = \boldsymbol{X}\bar{\boldsymbol{f}} .
\end{aligned} \tag{1}$$

Here $\bar{\boldsymbol{f}}$, $\bar{\boldsymbol{p}}$ and $\bar{\boldsymbol{x}}$ are vectors of applied, internal and redundant forces, respectively; $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{v}}$ are the vectors of node displacements and internal deformations work-conjugate to $\boldsymbol{f}$ and $\bar{\boldsymbol{p}}$, respectively; $\boldsymbol{B}_0$ and $\boldsymbol{B}_1$ are matrices that decompose the internal forces into statically determinate and indeterminate components, respectively, and $\boldsymbol{X}$ is a matrix relating redundants to applied forces. In the constitutive relation (1b), $\bar{\boldsymbol{F}}$ denotes the block-diagonal deformational flexibility matrix

$$\bar{\boldsymbol{F}} = \mathbf{diag}(\bar{\boldsymbol{F}}^{e}) \tag{2}$$

in which $\bar{\boldsymbol{F}}^{e}$ is the deformational-flexibility matrix of the $e$th finite element or substructure, called the compliance by some authors. A superposed bar is used to distinguish this classical flexibility, which plays no role in the present DFM, from the free–free flexibility matrix $\boldsymbol{F}$ introduced later. It should be noted that the deformation flexibility $\bar{\boldsymbol{F}}$ is required to be non-singular.

In Eqs. (1) we have largely followed the format of Pestel and Leckie [30] wherein subscripts 0 and 1 refer to determinate and indeterminate portions of the problem, respectively. If the structure is statically determinate, $\boldsymbol{B}_1$, $\boldsymbol{X}$ and $\bar{\boldsymbol{x}}$ are void, and (1a) suffices for the analysis.

The main decision for carrying out this method is the selection of redundants $\bar{\boldsymbol{x}}$, because all other steps are thereby determined. Matrices $\boldsymbol{B}_0$ and $\boldsymbol{B}_1$ are constructed through a variety of techniques driven by considerations discussed below. Matrices $\boldsymbol{D}_{00} = \boldsymbol{B}_0^{\mathrm{T}}\bar{\boldsymbol{F}}\boldsymbol{B}_0$, $\boldsymbol{D}_{10} = \boldsymbol{B}_1^{\mathrm{T}}\bar{\boldsymbol{F}}\boldsymbol{B}_0 = \boldsymbol{D}_{01}^{\mathrm{T}}$ and $\boldsymbol{D}_{11} = \boldsymbol{B}_1^{\mathrm{T}}\bar{\boldsymbol{F}}\boldsymbol{B}_1$ are then computed. The redundant forces, internal forces and node displacements are obtained *in tandem* from

$$\begin{aligned}
&\bar{\boldsymbol{x}} = \boldsymbol{X}\bar{\boldsymbol{f}} = -\boldsymbol{D}_{11}^{-1}\boldsymbol{D}_{10}\bar{\boldsymbol{f}} , \\
&\bar{\boldsymbol{p}} = \boldsymbol{B}_0\boldsymbol{f} + \boldsymbol{B}_1\boldsymbol{x} = (\boldsymbol{B}_0 + \boldsymbol{B}_1\boldsymbol{X})\bar{\boldsymbol{f}} , \\
&\bar{\boldsymbol{u}} = (\boldsymbol{B}_0 + \boldsymbol{B}_1\boldsymbol{X})^{\mathrm{T}}\bar{\boldsymbol{F}}\bar{\boldsymbol{p}} = (\boldsymbol{D}_{00} - \boldsymbol{D}_{10}^{\mathrm{T}}\boldsymbol{D}_{11}^{-1}\boldsymbol{D}_{10})\bar{\boldsymbol{f}} = \boldsymbol{F}_g\bar{\boldsymbol{f}} .
\end{aligned} \tag{3}$$

Matrix $\boldsymbol{D}_{11}$ must be non-singular in order for the global flexibility matrix $\boldsymbol{F}_g$, which is the inverse of the global stiffness matrix $\boldsymbol{K}_g$, to be uniquely determined. It is this non-singularity requirement that leads to the loss of locality and sparsity, a property enjoyed by the DSM. Only a minor part of the computations (3) can be carried out element by element, thus making the CFM non-competitive against the DSM.

Ideally, the choice of $\bar{\boldsymbol{x}}$ should yield a *well conditioned* and sparse $\boldsymbol{B}_1$ matrix. Those attributes are inherited by $\boldsymbol{D}_{11} = \boldsymbol{B}_1^{\mathrm{T}}\bar{\boldsymbol{F}}\boldsymbol{B}_1$, which is the coefficient matrix in the computationally dominant solution step $\boldsymbol{D}_{11}\boldsymbol{x} = -\boldsymbol{D}_{10}\bar{\boldsymbol{f}}$. For relatively simple truss and framework structures the hand selection of good redundants is well understood after decades of experience. As an illustration, in the four-bay Union-Jack plane truss shown in Fig. 1(a) four diagonal members are cut as depicted in Fig. 1(b), and their internal axial forces chosen as redundants $x_1$ through $x_4$. This is known to be a good selection because the redundants are strongly linearly independent, which helps numerical condition, and the effect of the force-pairs $x_i$ is localized, which helps sparsity. Note, however, that the effect of applied forces on the statically determinate subproblem (the truss upon removal of the four redundant members) is not necessarily localized. For example, the top force $F$ shown in Fig. 1(c) stresses almost all members. This is a consequence of the requirement of statical determinancy: any applied force must reach the supports through *load paths* that traverse that structure.

Human selection of good redundants becomes progressively difficult, however, as skeletal structures increase in complexity. And it would be unthinkable for the discrete models of plates, shells and three-dimensional

Table 1
Matrix notational conventions

| Symbol | Description |
|---|---|
| $\boldsymbol{B}_0, \boldsymbol{B}_1$ | Load-influence and self-strain matrices in CFM |
| $\boldsymbol{C}, \boldsymbol{H}$ | Auxiliary matrices in computation of free–free flexibility |
| $\boldsymbol{D}_{00}, \ldots, \boldsymbol{D}_{11}$ | Solution submatrices in CFM |
| $\boldsymbol{D}_f$ | Diagonal matrix $\boldsymbol{L}^T\boldsymbol{L}$ |
| $\overline{\boldsymbol{F}}$ | Block diagonal substructural flexibility |
| $\overline{\boldsymbol{F}}$ | Deformational flexibility matrix in CFM |
| $\boldsymbol{F}_g$ | Flexibility of the total assembled structure |
| $\overline{\boldsymbol{K}}$ | Block diagonal substructural stiffness |
| $\boldsymbol{K}_g$ | Stiffness of the total assembled structure |
| $\boldsymbol{I}$ | Identity matrix ($\boldsymbol{I}_k$: $k \times k$ identity matrix) |
| $\boldsymbol{G}$ | Globalization matrix: generalized inverse of $\boldsymbol{L}$ |
| $\boldsymbol{L}$ | Localization matrix linking local and global freedoms |
| $\boldsymbol{P}_r$ | Rigid-body mode projector |
| $\boldsymbol{P}_f$ | Projector for $\boldsymbol{R}^T\boldsymbol{L}$ |
| $\boldsymbol{R}$ | Substructural rigid-mode matrix |
| $\boldsymbol{T}$ | Transformation matrix in computation of free–free flexibility |
| $\boldsymbol{U}$ | Boundary localization matrix |
| $\boldsymbol{X}$ | Redundant influence matrix in CFM |
| $\boldsymbol{W}, \boldsymbol{Y}$ | Symmetric matrices appearing in iterative solution |
| $\boldsymbol{0}$ | Null matrix or vector |
| | |
| $\boldsymbol{d}$ | Deformational displacements |
| $\hat{\boldsymbol{d}}$ | Known portion of deformational displacements |
| $\boldsymbol{p}$ | Internal forces for all substructures |
| $\boldsymbol{f}$ | Applied forces for all substructures |
| $\overline{\boldsymbol{f}}$ | Applied forces in the CFM |
| $\boldsymbol{f}_g$ | Globally (assembled) applied force |
| $\boldsymbol{r}_\lambda, \boldsymbol{r}_r, \boldsymbol{r}$ | Interface residuals in iterative solution |
| $\boldsymbol{u}$ | Total substructural displacements |
| $\overline{\boldsymbol{u}}$ | Total node displacements in the CFM |
| $\boldsymbol{u}_r$ | Rigid body mode displacements |
| $\boldsymbol{u}_g$ | Total interface displacement at global level |
| $\overline{\boldsymbol{v}}$ | Generalized deformations in CFM |
| $\overline{\boldsymbol{x}}$ | Redundants in CFM |
| $\boldsymbol{\alpha}$ | Rigid body mode amplitudes for all substructures |
| $\boldsymbol{\lambda}_b$ | Interface interaction forces for all substructures |
| | |
| $(\cdot)^e$ | Pertaining to individual element ($e$) |
| $(\cdot)^s$ | Pertaining to individual substructure ($s$) |
| $(\cdot)_n$ | Pertaining to node $n$ |
| $(\cdot)_b$ | Pertaining to substructure boundary node freedoms |
| $(\cdot)_i$ | Pertaining to substructure internal node freedoms |
| $(\cdot)_f$ | Pertaining to substructure boundary force-specified freedoms |
| $(\cdot)_d$ | Pertaining to substructure boundary displacement-specified freedoms |

solids. By 1960 it was evident that if the CFM was to compete against the up-and-coming, fully automatic Direct Stiffness Method [31–33], a computerized selection of $\overline{\boldsymbol{x}}$ was required. Procedures informally known as 'structural cutters' were developed [16,17]. The best ones operated by directly forming $\boldsymbol{B}_1$ column by column without bothering about the physical interpretation of the $\overline{\boldsymbol{x}}$. Strong linear independence of the columns was enforced through pivoting techniques related to Gauss–Jordan elimination. Unfortunately, this strategy generally results in a dense $\boldsymbol{B}_1$, destroying the sparsity of $\boldsymbol{D}_{11}$. When these procedures were tried on continuum models, large increases in storage and solution costs with respect to the DSM were observed.

Loss of sparsity is not the only drawback. The CFM is based on equilibrium, constitutive and compatibility relations expressed in a *node-to-node* basis. Linkage of node to element properties presents no problems in bars and beams. For instance, the discrete constitutive equations $\overline{\boldsymbol{v}}^s = \overline{\boldsymbol{F}}^s \overline{\boldsymbol{p}}^s$ of a two-node beam element are readily constructed, since the relative displacement and rotations of the end nodes may be chosen as deformation

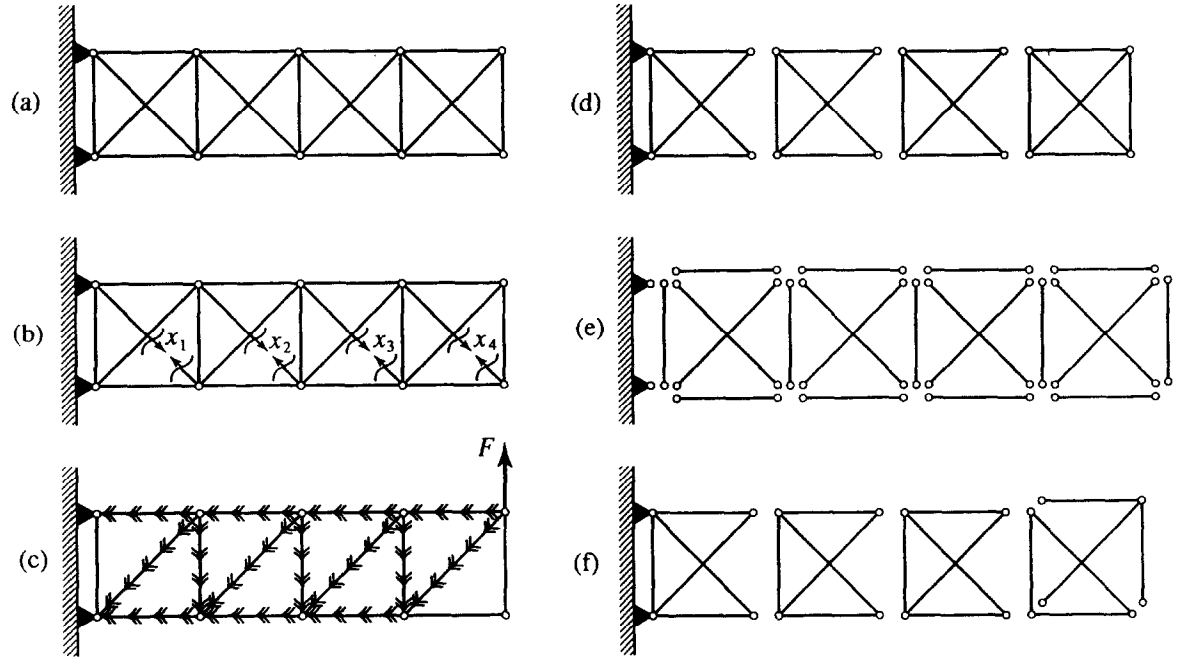


Fig. 1. (a) A four-bay, statically indeterminate plane truss; (b), (c): treatment by the Classical Force Method; (d)–(f): partitioned structures for the Direct Flexibility Method. The latter are discussed in Section 3.

variables $\bar{\boldsymbol{v}}^s$ conjugate to the standard internal force resultants (axial forces, shears and moments) in $\bar{\boldsymbol{p}}^s$. Then $\bar{\boldsymbol{p}}^s$ and $\bar{\boldsymbol{v}}^s$ can be easily linked to nodal quantities to establish a split system of equilibrium and compatibility equations.

Serious difficulties arise, however, in continuum models containing plates, shells or solid elements. The task becomes nontrivial in simple triangle and tetrahedron elements, and exceedingly hard in more complex ones. For example, consider a 20-node, 60-dof curved isoparametric brick: exactly 54 independent stress patterns must be chosen as internal forces along with 54 conjugate strains, and these linked to nodal forces and displacements. This is a formidable combinatorial problem. Now consider an internal node where eight such bricks meet. The nodal quantities will depend on $8 \times 54 = 432$ internal force patterns, which in turn depend on all connecting node variables. A 'structure cutter' traversing such a maze has negligible chance of finding a sparse basis.

The foregoing review indicates that the fundamental CFM building block that consists of a nodally-connected, statically-determinate subproblem modified by a set of redundancies, gives rise to serious computational difficulties when applied to continuum models. These difficulties explain the disappearance of the CFM from general-purpose FE codes by 1970, and have motivated us to formulate the Direct Flexibility Method (DFM) along a different track. This method emphasizes substructural computations realized through appropriate model decompositions. In other words, the loss of locality and sparsity inherent in the CFM is obviated by carrying out the computations on disconnected substructures. In so doing one also automatically benefits from *locality* of force distributions. Locality improves sparsity and exploitation of parallelism, and facilitates the treatment of inverse problems.

## 3. DFM Step 1: Model decomposition

The main steps of the present DFM are illustrated in Fig. 2. This section focuses on the first step. The original finite element model is decomposed into nonoverlapping substructures satisfying certain rank requirements discussed below. A substructure may reduce to an individual finite element as a special case. This configuration is called the *partitioned structure*. In nonstructural applications of the DFM, the term *partitioned model* may be

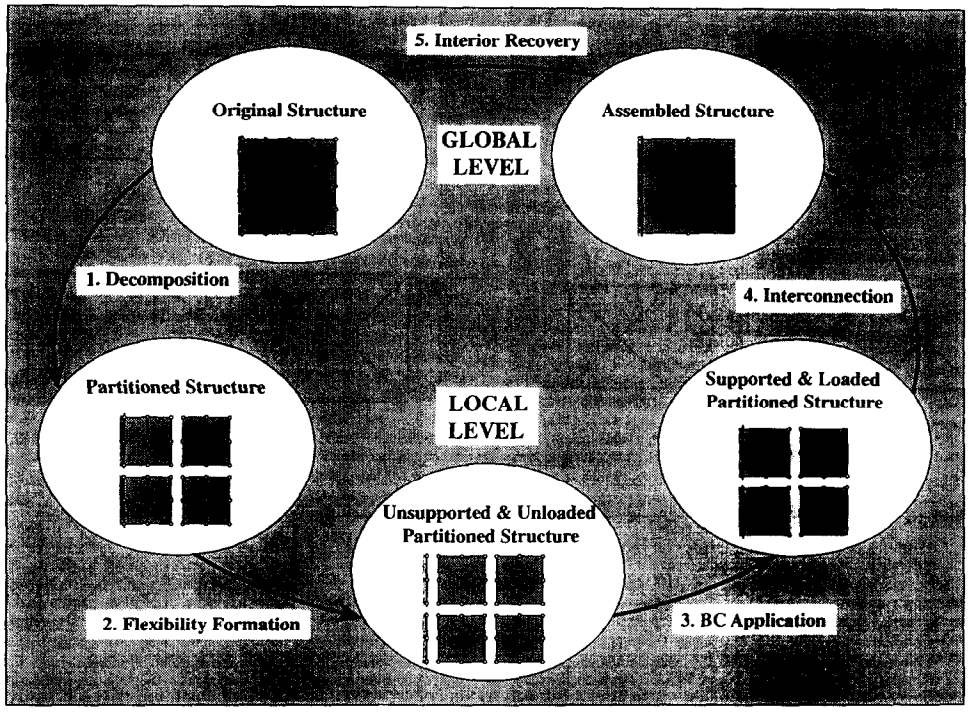


Fig. 2. The main steps of the DFM. Thumbnail FE pictures pertain to the plane stress mesh shown in more detail in Figs. 3 and 4.

used. Readers familiar with partitioned analysis concepts will observe that the DFM adopts element-by-element partitions, rather than node-by-node partitions characteristic of the CFM.

Each node of the partitioned structure is assigned a nonnegative integer attribute $V$ called its *valency*. If the node is located on the boundary of one or more substructures, $V$ counts the number of substructures it belong to. Otherwise $V$ is zero. With the help of this counter nodes are classified into three types:

$$\text{Boundary nodes} \begin{cases} \text{Interface nodes: } V \geqslant 2\,. \\ \text{External nodes: } V = 1\,. \end{cases}$$

$$\text{Internal nodes: } V = 0\,.$$

Node-resident quantities such as forces and displacements are distinguished by the same qualifiers; for example *interface forces* are only defined at nodes shared by two or more substructures. The term *cross nodes* is used in the original FETI method [28] to identify those with $V \geqslant 3$.

A substructure may possess $N_r > 0$ rigid-body modes when partitioned as shown in Fig. 2. If so it is called a *floating* substructure. If no rigid-body mode is suppressed the substructure is called *free–free*. If all of them are suppressed ($N_r = 0$) through appropriate support conditions, it is called *fixed*.

The decomposition step is best illustrated through examples. Fig. 1(d) through 1(f) show three partitioned configurations of the plane truss of Fig. 1(a). The partitioned structure of Fig. 1(d) results from a decomposition into four substructures, of which three are free–free (with $N_r = 3$) and one fixed. Note that the rightmost substructure is statically indeterminate. Fig. 1(e) continues the decomposition down to 21 individual elements, all of which are free–free. The decomposition of Fig. 1(f) gives rise to a mechanism due to the rightmost substructure. Partitioning that gives rise to mechanisms requires special treatment, which will not be considered in this paper.

Fig. 3 illustrates the decomposition of a continuum model. Fig. 3(a) shows a square plane stress structure clamped along $AB$, which is discretized by a regular $4 \times 4$ mesh of four-node elements. Elements are identified by numbers (1) through (16) while nodes are identified by *global* numbers 1 to 25. Fig. 3(c) shows a generic element ($e$) and its *local* node numbering.

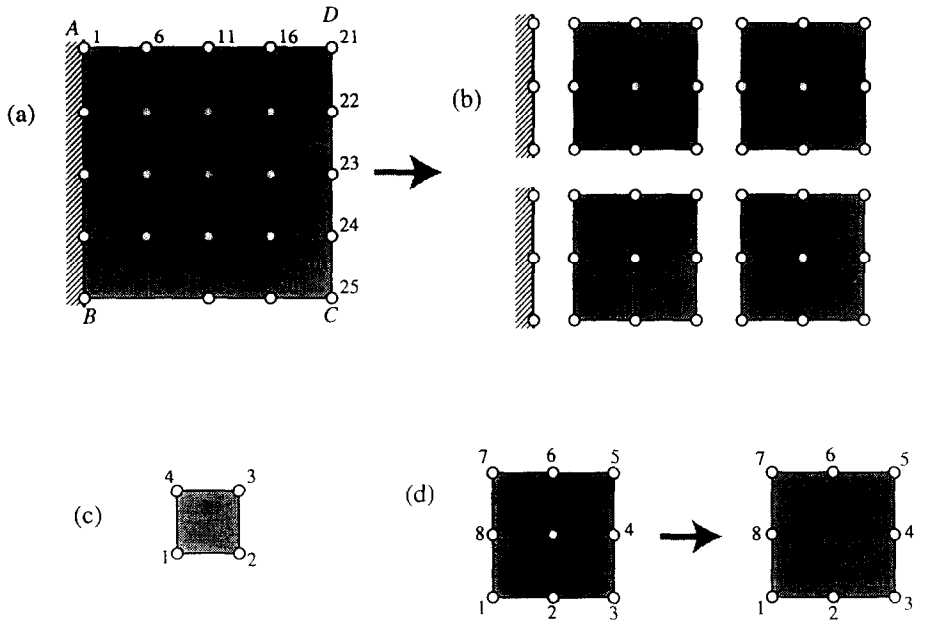


Fig. 3. (a) A regular $4 \times 4$ plane-stress mesh used as example in Sections 3–7 wheres (b) is a decomposition of (a) into four substructures. (c) shows a generic element ($e$) of the original structure, and (d) a generic substructure ($s$) before and after removal of its interior nodes.

Fig. 3(b) depicts a partitioned structure consisting of four identical $4 \times 4$ disconnected substructures, identified as (1) through (4). In this case two substructures are free–free (with $N_r = 3$) whereas two are fixed. Fig. 3(d) displays a generic substructure ($s$) before and after the interior nodes are eliminated as discussed in Step 4. The substructure nodes are identified as shown; for convenience the interior nodes are numbered last. Note that none of the decompositions illustrated in Figs. 1 and 3 would be acceptable for the CFM because the partitioned structures are mechanisms.

## 4. Step 2: Flexibility formation

The second DFM step involves the formation of the matrix flexibility equations for the partitioned substructures. On completion only the substructure boundary nodes remain, as illustrated in Fig. 3(d). The bulk of the work involves the formation of the substructure flexibility matrices. On distributed-memory parallel computers, this work is trivially task-parallelizable if each substructure is assigned to a separate processor.

Consider an individual substructure ($s$) including internal nodes. The substructure is made free–free by replacing any supports by reaction node forces as necessary. and including these reactions in the interior node forces. The total number of nodal degrees of freedom is called $N_f$.

Being free–free, the substructure has $N_r^s$ unsuppressed rigid body modes (RBM). Let a linearly independent modal basis for the RBM be chosen as columns of a matrix $\boldsymbol{R}^s$ dimensioned $N_f^s \times N_r^s$, so that the rigid body nodal displacements can be represented as $\boldsymbol{u}_r^s = \boldsymbol{R}^s \boldsymbol{\alpha}^s$, where $\boldsymbol{\alpha}^s$ is a column vector containing $N_r^s$ RBM amplitudes. For computational convenience the columns are orthonormalized to satisfy $(\boldsymbol{R}^s)^{\mathrm{T}} \boldsymbol{R}^s = \boldsymbol{I}$. The direct construction of $\boldsymbol{R}^s$ by geometric arguments is explained in Section 9.

We often use in the sequel the *orthogonal RBM projector*

$$\boldsymbol{P}_r^s = \boldsymbol{I} - \boldsymbol{R}^s (\boldsymbol{R}^s)^{\mathrm{T}}, \tag{4}$$

which is a symmetric and idempotent matrix: $(\boldsymbol{P}_r^s)^2 = \boldsymbol{P}_r^s$.

For each substructure ($s$), the force equilibrium and the displacement decomposition can be expressed as (see Fig. 4):

$$\boldsymbol{p}^s = \boldsymbol{f}^s + \boldsymbol{U}^s \boldsymbol{\lambda}_b^s, \qquad \boldsymbol{u}^s = \boldsymbol{d}^s + \boldsymbol{r}^s = \boldsymbol{d}^s + \boldsymbol{R}^s \boldsymbol{\alpha}^s, \tag{5}$$

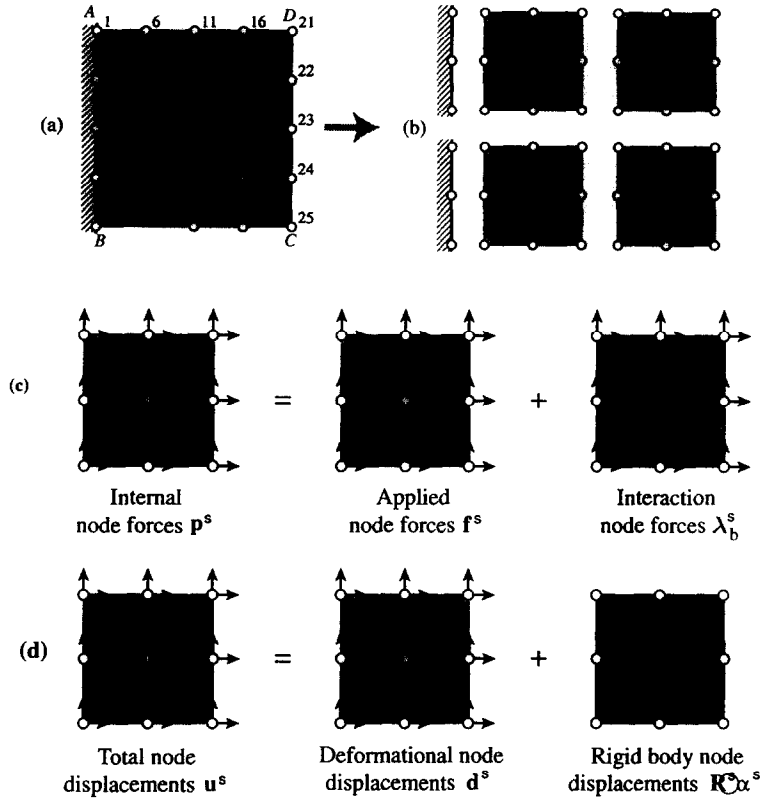


Fig. 4. The composition of node force and displacement quantities for an individual substructure: (c) equilibrium, (d) kinematics.

where $\boldsymbol{\lambda}_b$ is the vector of interface nodal forces, and $\boldsymbol{U}^s$ is a Boolean matrix whose entries are one and zero for substructural interior and interface nodes, respectively. For example, for the case of a substructure made of four plane stress elements as shown in Fig. 3(d), $\boldsymbol{U}^s$ becomes an $(18 \times 18)$ diagonal matrix whose first 16 entries are unity and the seventeen and eighteen entries are zero as the interior node number is 9.

Eq. (5a) states that the substructural internal node forces $\boldsymbol{p}^s$ must be in equilibrium with the substructural applied forces $\boldsymbol{f}^s$ and the interface forces $\boldsymbol{\lambda}_b^s$ acting on the interface boundary. Eq. (5b) states that the nodal displacements $\boldsymbol{u}^s$ are the superposition of deformational displacements $\boldsymbol{d}^s = \boldsymbol{P}_r^s \boldsymbol{u}^s$ and rigid displacements $\boldsymbol{r}^s = (\boldsymbol{I} - \boldsymbol{P}_r^s)\boldsymbol{u}^s = \boldsymbol{R}^s \boldsymbol{\alpha}^s$. Note that $\boldsymbol{d}^s$ and $\boldsymbol{r}^s$ are orthogonal because $(\boldsymbol{I} - \boldsymbol{P}_r^s)\boldsymbol{P}_r^s = 0$.

From the Principle of Virtual Work, the self-equilibrium of the substructure ($s$) can be mathematically expressed as

$$(\boldsymbol{R}^s)^{\mathrm{T}} \boldsymbol{p}^s = (\boldsymbol{R}^s)^{\mathrm{T}} (\boldsymbol{f}^s + \boldsymbol{U}^s \boldsymbol{\lambda}_b^s) = \boldsymbol{0}\,, \tag{6}$$

which implies that the sum of forces and moments for each substructure must be zero.

We now introduce the dual relationships between the substructural stiffness $\boldsymbol{K}^s$ and substructural flexibility $\boldsymbol{F}^s$:

$$\boldsymbol{K}^s \boldsymbol{u}^s = \boldsymbol{p}^s = \boldsymbol{f}^s + \boldsymbol{U}^s \boldsymbol{\lambda}_b^s\,, \qquad \boldsymbol{F}^s \boldsymbol{p}^s = \boldsymbol{d}^s = \boldsymbol{u}^s - \boldsymbol{R}^s \boldsymbol{\alpha}^s\,. \tag{7}$$

The stiffness equation (7a) for a partitioned substructure is well known. $\boldsymbol{K}^s$ may be generated by standard DSM assembly techniques. Because of the ready availability of element stiffness routines, it is assumed that $\boldsymbol{K}^s$ will be constructed first. On the other hand, the flexibility equation (7b) employs the so-called *free–free substructural flexibility* matrix $\boldsymbol{F}^s$, which represents a generalization of the classical deformational flexibility matrix. Specifically, $\boldsymbol{K}^s$ and $\boldsymbol{F}^s$ are the Moore–Penrose generalized inverses of each other given by

$$\begin{aligned} &\boldsymbol{F}^s = \boldsymbol{P}_r^s [\boldsymbol{K}^s + \boldsymbol{R}^s (\boldsymbol{R}^s)^{\mathrm{T}}]^{-1} = (\boldsymbol{F}^s)^{\mathrm{T}}\,, \qquad \boldsymbol{K}^s = \boldsymbol{P}_r^s [\boldsymbol{F}^s + \boldsymbol{R}^s (\boldsymbol{R}^s)^{\mathrm{T}}]^{-1} = (\boldsymbol{K}^s)^{\mathrm{T}}\,, \\ &\boldsymbol{K}^s \boldsymbol{F}^s = \boldsymbol{F}^s \boldsymbol{K}^s = \boldsymbol{P}_r^s\,, \qquad \boldsymbol{K}^s \boldsymbol{P}_r^s = \boldsymbol{F}^s \boldsymbol{P}_r^s = \boldsymbol{0}\,. \end{aligned} \tag{8}$$

The most important property is that $\boldsymbol{F}^s$ and $\boldsymbol{K}^s$ share the same eigenvector basis. Of the infinite number of generalized inverses of a singular $\boldsymbol{K}^s$, this is the only one enjoying that spectral property. If the substructure possesses no RBMs, $\boldsymbol{R}^s$ is void, $\boldsymbol{P}_r^s = \boldsymbol{I}$, and $\boldsymbol{F}^s$ and $\boldsymbol{K}^s$ become the ordinary inverses of each other. The efficient computation of $\boldsymbol{F}^s$ is discussed in Section 9.

Having formed $\boldsymbol{K}^s$ by the DSM and $\boldsymbol{R}^s$ geometrically, $\boldsymbol{F}^s$ may be evaluated directly using the first of (8) if the substructure contains a few elements, or is an individual element. For substructures containing hundred or thousands of elements, exploitation of the natural sparseness of $\boldsymbol{K}^s$ becomes important, and a procedure to that effect is discussed in a separate article [34].

## 5. DFM Step 3: Application of kinematic boundary conditions

This step brings into play nodal displacement boundary conditions acting on an individual substructure $s$. A straightforward stiffness condensation method is presented here. Consider the following splitting of the first of (7):

$$\boldsymbol{K}^s\boldsymbol{u}^s = \boldsymbol{f}^s + \boldsymbol{U}^s\boldsymbol{\lambda}_b \Rightarrow \begin{bmatrix} \boldsymbol{K}_{ff}^s & \boldsymbol{K}_{fd}^s \\ \boldsymbol{K}_{df}^s & \boldsymbol{K}_{dd}^s \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_f^s \\ \boldsymbol{u}_d^s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_f^s + \boldsymbol{U}_f^s\boldsymbol{\lambda}_f^s \\ \boldsymbol{f}_d^s + \boldsymbol{U}_d^s\boldsymbol{\lambda}_d^s \end{Bmatrix}, \tag{9}$$

in which subscripts $f$ and $d$ denote substructural freedoms where forces and displacements are specified, respectively. Suppose that $\boldsymbol{u}_d^s = \hat{\boldsymbol{u}}_d^s$ because of support conditions which as a special case includes $\hat{\boldsymbol{u}}_d = \boldsymbol{0}$. The first row of (9) now becomes

$$\boldsymbol{K}_{ff}\boldsymbol{u}_f = \boldsymbol{f}_f + \boldsymbol{U}_f\boldsymbol{\lambda}_f - \boldsymbol{K}_{fd}\hat{\boldsymbol{u}}_d\,. \tag{10}$$

If we assume that the support conditions *preclude all substructure rigid body modes*, i.e. $\boldsymbol{u}_f = \boldsymbol{d}_f$:

$$\boldsymbol{F}_{ff}^s\boldsymbol{p}_f^s = \boldsymbol{u}_f^s + (\boldsymbol{K}_{ff}^s)^{-1}\boldsymbol{K}_{fd}^s\hat{\boldsymbol{u}}_d^s \;\Rightarrow\; \boldsymbol{F}^s\boldsymbol{p}^s = \boldsymbol{d}^s + \hat{\boldsymbol{d}}^s\,, \quad \boldsymbol{F}^s = \boldsymbol{K}_{ff}^{-1} \tag{11}$$

where the symbol ($\Rightarrow$) implies that the substructural flexibility $\boldsymbol{F}^s$ is the ordinary inverse of the substructural stiffness matrix corresponding to the force-specified nodes. Likewise, the substructural displacements become the substructural deformations $\boldsymbol{d}^s$. Matrices $\boldsymbol{F}^e$, $\boldsymbol{R}^s$ and the known displacement portion $\hat{\boldsymbol{d}}^s$ are carried forward into the next step. In passing we note that the second row of (9) may be used to compute the boundary reaction forces $\boldsymbol{\lambda}_d$.

As further discussed in Section 11, the case where support conditions suppress only a subset of the rigid body modes, leaving a floating substructure, is numerically difficult as it involves rank detection and the use of a generalized inverse in (11). A penalty approach that uses rigid (zero flexibility) boundary elements defers such modifications to subsequent steps and promises to overcome the problem of floating substructures. This method is not presented here because its efficient implementation is still under investigation.

## 6. Preliminaries for interconnection operations

The key data emerging from Step 3 is an array of matrices for the disconnected substructures:

$$\boldsymbol{F}^s, \boldsymbol{R}^s\,, \quad s = 1, 2, \ldots, N_s\,,$$

plus known vectors such as $\hat{\boldsymbol{d}}^s$ and $\boldsymbol{f}^s$. In preparation for the interconnection calculations discussed in Section 7, we summarize here the quantities and relations required therein. The partitioned structure of Fig. 3 illustrates the definition of matrix and vectors. The assembled structure in Fig. 2 shows the interface nodes (18 local, 8 global), which are left over after application of known force and displacement conditions. Those are used to define the following matrix and vector entities.

### 6.1. Assembly matrix

First, we introduce a localization matrix $\boldsymbol{L}$ that relates the assembled global nodal displacements $\boldsymbol{u}_g$ to the partitioned substructural displacements (including elements) $\boldsymbol{u}^s$ as

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}^1 \\ \boldsymbol{u}^2 \\ \vdots \\ \boldsymbol{u}^{N_s} \end{bmatrix} = \boldsymbol{L}\boldsymbol{u}_g \,. \tag{12}$$

Here, $\boldsymbol{u}^s$ include all degrees of freedom (interior and interface) of substructure ($s$). The localization matrix $\boldsymbol{L}$ is split into the substructural interior part $\boldsymbol{L}_i$ and substructural boundary node part $\boldsymbol{L}_b$:

$$\boldsymbol{L} = \begin{bmatrix} \boldsymbol{L}_i & 0 \\ 0 & \boldsymbol{L}_b \end{bmatrix} \tag{13}$$

Because the substructural interior nodes do not appear in the interconnection conditions, we focus on the boundary localization matrix $\boldsymbol{L}_b$. The matrix relates local to global interface freedoms only. The $(i, j)$ entry is one if $i$th-local freedom links to the $j$th global freedom, and zero otherwise. For the example structure shown at the lower right corner in Fig. 2, this matrix is $36 \times 16$. To save space a node-by-node version of $\boldsymbol{L}^{\mathrm{T}}$ is shown below, with $\boldsymbol{I}$ and $\boldsymbol{0}$ denoting the $2 \times 2$ identity and zero matrices, respectively:

$$\boldsymbol{L}_b^{\mathrm{T}} = [\quad \boldsymbol{L}^1 \qquad\qquad \boldsymbol{L}^2 \qquad\qquad \boldsymbol{L}^3 \qquad\qquad\qquad \boldsymbol{L}^4 \qquad\qquad ]^{\mathrm{T}}$$

$$= \begin{bmatrix}
\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0}
\end{bmatrix}. \tag{14}$$

Extra spaces group the columns of $\boldsymbol{L}_b^{\mathrm{T}}$ into contributions from the four substructures. In practice this matrix can be represented compactly by a pointer array. The product $\boldsymbol{D}_l = \boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b$ is a diagonal matrix:

$$\boldsymbol{D}_l = \boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b = \mathbf{diag}(2, 2, 2, 2, 2, 2, 4, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2) \,. \tag{15}$$

The diagonal value, repeated for the $x$ and $y$ freedoms, counts how many substructures meet at a node. That count was called the nodal valency $V$ in Section 3. The above matrix appears in the expression of the globalization matrix $\boldsymbol{G}_b$, which is the generalized Penrose left-inverse of $\boldsymbol{L}_b$:

$$\boldsymbol{G}_b = \boldsymbol{L}_b(\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{L}_b)^{-1} = \boldsymbol{L}_b(\boldsymbol{D}_l)^{-1} \,, \qquad \boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{G}_b = \boldsymbol{I} \,. \tag{16}$$

This is identical in configuration to $\boldsymbol{L}_b$ except that each 1-entry changes to $1/V$.

In order to obtain the interface flexibility matrices, we first construct the substructural flexibility matrix for each substructure as given by (8). Then referring to Fig. 3, we extract the interface flexibility matrix of all the substructures by using the following formula:

$$\boldsymbol{F}_b = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{U} \,, \qquad \boldsymbol{U} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I}_b \end{bmatrix}, \qquad \boldsymbol{F} = \begin{bmatrix} \boldsymbol{F}^1 & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{F}^2 & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}^3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{F}^4 \end{bmatrix} \tag{17}$$

where $\boldsymbol{I}_b$ is the boundary-node Boolean operator of order $36 \times 36$ for the example problem pertaining to the matrix $\boldsymbol{L}_b$, and the substructural flexibility matrices $\boldsymbol{F}^1$, $\boldsymbol{F}^2$, $\boldsymbol{F}^3$ and $\boldsymbol{F}^4$ correspond to the partitioned case of Fig. 5(a). Note that $\boldsymbol{F}^1$ and $\boldsymbol{F}^2$ come from the displacement boundary-treated flexibility (11) whereas $\boldsymbol{F}^3$ and $\boldsymbol{F}^4$ come from the free–free flexibility formula (8).

Similarly, the substructural interface rigid-body modes can be obtained as

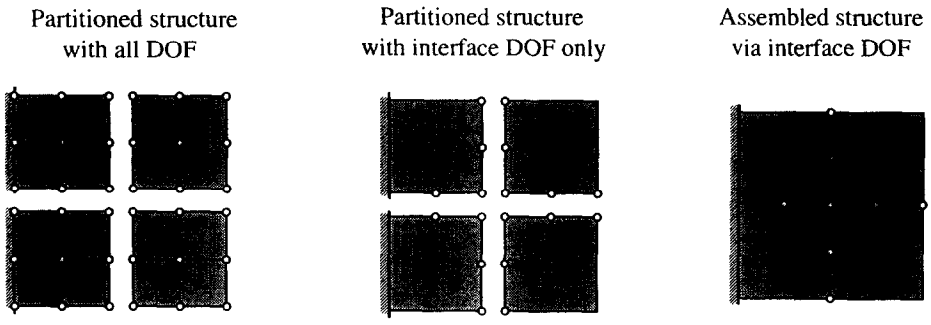


Fig. 5. The partitioned structure of Fig. 3 showing the total substructural nodes, and the interface nodes that will appear in the interconnection calculations.

$$\boldsymbol{R}_b = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{R}\,, \qquad \boldsymbol{R} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{R}^3 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{R}^4 \end{bmatrix}, \tag{18}$$

where $\boldsymbol{R}^1$ and $\boldsymbol{R}^2$ are empty because substructures 1 and 2 are fully supported, and the sizes of $\boldsymbol{R}^3$ and $\boldsymbol{R}^4$ are $18 \times 3$.

### 6.2. Substructural and global vectors

Six substructural column vectors appear in the interconnection step: $\boldsymbol{\alpha}$, $\boldsymbol{d}$, $\boldsymbol{f}$, $\boldsymbol{\lambda}_b$, $\boldsymbol{p}$ and $\boldsymbol{u}$. The common order of the last five vectors is denoted by $N_l$ and that of $\boldsymbol{\alpha}$ by $N_r$. For example, $\boldsymbol{d}$, $\boldsymbol{\lambda}_b$, and $\boldsymbol{\alpha}$ represent the total substructural quantities:

$$\boldsymbol{d} = \begin{bmatrix} \boldsymbol{d}^1 \\ \boldsymbol{d}^2 \\ \boldsymbol{d}^3 \\ \boldsymbol{d}^4 \end{bmatrix}, \qquad \boldsymbol{\lambda}_b = \begin{bmatrix} \boldsymbol{\lambda}_b^1 \\ \boldsymbol{\lambda}_b^2 \\ \boldsymbol{\lambda}_b^3 \\ \boldsymbol{\lambda}_b^4 \end{bmatrix}, \qquad \boldsymbol{\alpha} = \begin{bmatrix} \boldsymbol{\alpha}^3 \\ \boldsymbol{\alpha}^4 \end{bmatrix} \tag{19}$$

the last from the fact substructures (1) and (2) do not have rigid-body modes. Vectors $\boldsymbol{p}$ and $\boldsymbol{u}$ are defined in the same way as the deformation vector $\boldsymbol{d}$. Two global vectors of forces and displacements appear: $\boldsymbol{f}_g$ and $\boldsymbol{u}_g$. The common order of $\boldsymbol{f}_g$ and $\boldsymbol{u}_g$ is denoted by $N_g$.

The linkage between the local and global forces is given by the dual of the displacement localization relation (12):

$$\boldsymbol{f}_g = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{f} \Leftrightarrow \boldsymbol{u} = \boldsymbol{L}\boldsymbol{u}_g\,. \tag{20}$$

### 6.3. Projectors

A rigid-body mode projector is defined using the block diagonal $\boldsymbol{R}_b$:

$$\boldsymbol{P}_{rb} = \boldsymbol{I} - \boldsymbol{R}_b(\boldsymbol{R}_b^{\mathrm{T}}\boldsymbol{R}_b)^{-1}\boldsymbol{R}_b^{\mathrm{T}} = \boldsymbol{P}_{rb}^{\mathrm{T}}\,. \tag{21}$$

This matrix needs not be explicitly constructed because it only appears in matrix–vector multiplies. Such operation: $\boldsymbol{z} = \boldsymbol{P}_{rb}\boldsymbol{y} = \boldsymbol{y} - \boldsymbol{R}_b(\boldsymbol{R}_b^{\mathrm{T}}\boldsymbol{R}_b)^{-1}\boldsymbol{R}_b^{\mathrm{T}}\boldsymbol{y}$ may be carried out locally, that is, in substructure-level computations.

In the iterative solution discussed in the next section, the following projector appears:

$$\boldsymbol{P}_l = \boldsymbol{I} - \boldsymbol{L}_r(\boldsymbol{L}_r^{\mathrm{T}}\boldsymbol{L}_r)^{-1}\boldsymbol{L}_r^{\mathrm{T}} \tag{22}$$

in which $\boldsymbol{L}_r = \boldsymbol{P}_{rb}\boldsymbol{L}_b$. Using the Woodbury formula, $(\boldsymbol{L}_r^{\mathrm{T}}\boldsymbol{L}_r)^{-1}$ can be obtained as follows:

$$\boldsymbol{W}^{-1} = (\boldsymbol{L}_r^{\mathrm{T}}\boldsymbol{L}_r)^{-1} = \boldsymbol{D}_l^{-1} + \boldsymbol{D}_l^{-1}\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{R}_b\boldsymbol{Y}^{-1}\boldsymbol{R}_b^{\mathrm{T}}\boldsymbol{L}_b\boldsymbol{D}_l^{-1}, \qquad \boldsymbol{Y} = \boldsymbol{R}_b^{\mathrm{T}}(\boldsymbol{I} - \boldsymbol{L}_b\boldsymbol{D}_l^{-1}\boldsymbol{L}_b^{\mathrm{T}})\boldsymbol{R}_b . \tag{23}$$

Because $\boldsymbol{D}_l$ is diagonal the only nontrivial factorization is now that of the $N_r \times N_r$ symmetric matrix $\boldsymbol{Y}$. $N_r$ is at most $3N_s$ in two dimensions and $6N_s$ in three. For example, if a $1000 \times 1000$ plane stress 2D mesh is partitioned into one hundred $100 \times 100$ substructures, matrix $\boldsymbol{W}$ is dimensioned $35\,802 \times 35\,802$ whereas $\boldsymbol{Y}$ is only $300 \times 300$. In massively parallel processing, the factorization of $\boldsymbol{Y}$, which is a symmetric sparse matrix, can be done in advance and then broadcast-copied to each processor.

## 7. DFM Step 4: Interconnection

The fourth DFM step involves the interconnection of the substructures thereby returning to the global level (cf. Fig. 2). The finite element interconnection equations are summarized below:

| | |
|---|---|
| Node by node equilibrium: | $\boldsymbol{p} = \boldsymbol{f} + \boldsymbol{U}\boldsymbol{\lambda}_b$, |
| Constitutive (flexibility form): | $\boldsymbol{F}\boldsymbol{p} = \boldsymbol{d} + \hat{\boldsymbol{d}} = \boldsymbol{u} - \boldsymbol{R}\boldsymbol{\alpha} + \hat{\boldsymbol{d}}$, |
| Substructure self-equilibrium: | $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{p} = \boldsymbol{0}$, |
| Interface compatibility: | $\boldsymbol{U}^{\mathrm{T}}(\boldsymbol{u} - \boldsymbol{L}\boldsymbol{u}_g) = 0$, |
| Interface equilibrium: | $\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{\lambda}_b = \boldsymbol{0}$. |

(24)

On eliminating $\boldsymbol{p}$, $\boldsymbol{d}$, $\boldsymbol{u}$, the foregoing equations yield the following symmetric coupled system in which $\boldsymbol{\lambda}_b$, $\boldsymbol{\alpha}$ and $\boldsymbol{u}_b$ are retained as unknowns:

$$\begin{bmatrix} \boldsymbol{F}_b & \boldsymbol{R}_b & -\boldsymbol{L}_b \\ \boldsymbol{R}_b^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{L}_b^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda}_b \\ \boldsymbol{\alpha} \\ \boldsymbol{u}_b \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{U}^{\mathrm{T}}(-\boldsymbol{F}\boldsymbol{f} + \hat{\boldsymbol{d}}) \\ -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{f} \\ \boldsymbol{0} \end{Bmatrix} . \tag{25}$$

Here, $\boldsymbol{u}_b = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{u}$ is the global displacement at the interface nodes and $\boldsymbol{F}_b = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{U}$ is the boundary flexibility.

This is the DFM interconnection equation. It may be solved by direct or iterative methods. The iterative solution is more interesting as ingredient of a scalable parallel solver for very large systems. Only a summary of the method is given here; more details may be found in a separate article [4]. The residual of the first matrix equation is

$$\boldsymbol{r}_\lambda = \boldsymbol{U}^{\mathrm{T}}(-\boldsymbol{F}\boldsymbol{f} + \hat{\boldsymbol{d}}) - \boldsymbol{F}_b\boldsymbol{\lambda}_b - \boldsymbol{R}_b\boldsymbol{\alpha} + \boldsymbol{L}_b\boldsymbol{u}_b , \tag{26}$$

which physically measures the interface compatibility violation. Vectors $\boldsymbol{\alpha}$ and $\boldsymbol{u}_b$ are eliminated by premultiplying $\boldsymbol{r}_\lambda$ by the projectors $\boldsymbol{P}_r$ and $\boldsymbol{P}_l$ defined in the previous section:

$$\boldsymbol{r} = \boldsymbol{P}_l\boldsymbol{P}_r\boldsymbol{r}_\lambda = \boldsymbol{P}_l\boldsymbol{P}_r[\boldsymbol{U}^{\mathrm{T}}(\hat{\boldsymbol{d}} - \boldsymbol{F}\boldsymbol{f}) - \boldsymbol{F}_b\boldsymbol{\lambda}_b] . \tag{27}$$

This projected residual is used with a preconditioned conjugate gradient (PCG) solver. The following stiffness preconditioner has given good results:

$$\boldsymbol{K}^{\mathrm{pre}} = \boldsymbol{P}_l\boldsymbol{P}_r\boldsymbol{K}_b\boldsymbol{P}_r\boldsymbol{P}_l . \tag{28}$$

Here, $\boldsymbol{K}_b$ denotes the block-diagonal matrix of Schur-complement boundary stiffnesses (see Section 9 for details):

$$\boldsymbol{K}_b = \mathbf{diag}(\boldsymbol{K}_b^s), \qquad \boldsymbol{K}_b^s = \boldsymbol{P}_r^s[\boldsymbol{P}_r^s\boldsymbol{F}_b^s\boldsymbol{P}_r^s + \boldsymbol{R}_b^s[(\boldsymbol{R}_b^s)^{\mathrm{T}}\boldsymbol{R}_b^s]^{-1}(\boldsymbol{R}_b^s)^{\mathrm{T}}]^{-1} . \tag{29}$$

All PCG steps can be carried out on a substructure by substructure basis except those involving $\boldsymbol{P}_l$. Matrix–vector operations with this projector, which represents the so-called 'coarse problem' can be streamlined through Eqs. (21)–(23), with $\boldsymbol{Y}$ prefactored before embarking in the PCG. Once $\boldsymbol{\lambda}_b$ has converged to the desired accuracy, the other quantities can be recovered from

$$\boldsymbol{u}_b = \boldsymbol{W}^{-1}\boldsymbol{L}_r^{\mathrm{T}}(\boldsymbol{U}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{f} + \hat{\boldsymbol{d}} - \boldsymbol{F}_b\boldsymbol{\lambda}_b), \qquad \boldsymbol{\alpha} = \boldsymbol{R}_b^{\mathrm{T}}(\hat{\boldsymbol{d}} + \boldsymbol{L}_b\boldsymbol{u}_b) , \tag{30}$$

in which for the former the efficient form of $\boldsymbol{W}^{-1}$ (23) should be used.

If a direct method is chosen to solve (25), elimination of $\boldsymbol{\lambda}_b$ and $\boldsymbol{\alpha}$ yields

$$\boldsymbol{L}_b^{\mathrm{T}}\boldsymbol{K}_b\boldsymbol{L}_b\boldsymbol{u}_g = \boldsymbol{f}_g\,, \tag{31}$$

This is the equation of the substructured form of the Direct Stiffness Method. If every substructure reduced to one element, the conventional DSM is recovered. Since these direct methods are known not to scale when used on massively parallel computers with hundreds or thousands of processors, they are of little practical interest compared to the iterative approach for such applications.

The last DFM step recovers the solution at non-interface nodes by standard back substitution techniques and need not be discussed here.

## 8. Variational formulation

The following material, extracted from [6], presents a variational framework for matrix flexibility methods and shows how the DFM fits in a class of such methods. To this end, we introduce the displacement-based discrete energy functional $J$ for a linear structure under quasistatic loads given by

$$J(\boldsymbol{u}_g) = \boldsymbol{u}_g^{\mathrm{T}}(\boldsymbol{f}_g - \tfrac{1}{2}\,\boldsymbol{K}_g\boldsymbol{u}_g)\,, \qquad \boldsymbol{K}_g = \boldsymbol{L}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{L}\,, \qquad \boldsymbol{K} = \mathbf{diag}(\boldsymbol{K}^s)\,. \tag{32}$$

where $\boldsymbol{K}$ is the block diagonal collection of unassembled substructural stiffness matrices.

Introducing the release $\boldsymbol{u}-\boldsymbol{L}\boldsymbol{u}_g$ of Eq. (12) through a Lagrange multiplier vector $\boldsymbol{\lambda}_b$ converts (32) to a three-variable functional

$$J(\boldsymbol{u}_g, \boldsymbol{\lambda}_b, \boldsymbol{u}) = \boldsymbol{u}_g^{\mathrm{T}}(\boldsymbol{L}^{\mathrm{T}}\boldsymbol{f} - \tfrac{1}{2}\,\boldsymbol{L}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{L}\boldsymbol{u}_g) = \boldsymbol{u}^{\mathrm{T}}\left(\boldsymbol{f} - \frac{1}{2}\,\boldsymbol{K}\boldsymbol{u}\right) + \boldsymbol{\lambda}_b^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{u} - \boldsymbol{L}\boldsymbol{u}_g)\,, \tag{33}$$

where $\boldsymbol{B}$ is a constraint weighting matrix to be determined. This can be further expanded by dividing the substructural displacements into deformational and rigid:

$$\boldsymbol{u} = \boldsymbol{d} + \boldsymbol{u}_r = \boldsymbol{d} + \boldsymbol{R}\boldsymbol{\alpha}\,, \tag{34}$$

which inserted into (33) yields a four-variable functional

$$J(\boldsymbol{\lambda}_b, \boldsymbol{\alpha}, \boldsymbol{d}, \boldsymbol{u}_g) = \boldsymbol{d}^{\mathrm{T}}(\boldsymbol{f} - \tfrac{1}{2}\,\boldsymbol{K}\boldsymbol{d}) + \boldsymbol{\lambda}_b^{\mathrm{T}}\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{d} - \boldsymbol{L}\boldsymbol{u}_g) + \boldsymbol{\lambda}_r^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}}(\boldsymbol{f} + \boldsymbol{B}\boldsymbol{\lambda}_b)\,. \tag{35}$$

The four state variables $(\boldsymbol{d}, \boldsymbol{\lambda}_b, \boldsymbol{\alpha}, \boldsymbol{u}_g)$ in the above equation are linearly independent provided that the constraint matrix $\boldsymbol{B}$ has full row rank. In other words, it is variationally complete. Under that condition the first variation is

$$\begin{aligned}\delta J = {} & \delta\boldsymbol{d}^{\mathrm{T}}(\boldsymbol{f} - \boldsymbol{K}\boldsymbol{d} + \boldsymbol{B}\boldsymbol{\lambda}_b) + \delta\boldsymbol{\lambda}_b^{\mathrm{T}}\,\boldsymbol{B}^{\mathrm{T}}(\boldsymbol{d} - \boldsymbol{L}\boldsymbol{u}_g + \boldsymbol{R}\boldsymbol{\alpha}) \\ & + \delta\boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{R}^{\mathrm{T}}(\boldsymbol{f} + \boldsymbol{B}\boldsymbol{\lambda}_b) - \delta\boldsymbol{u}_g^{\mathrm{T}}\,\boldsymbol{L}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{\lambda}_b\,.\end{aligned} \tag{36}$$

The stationarity condition $\delta J = 0$ yields

$$\begin{bmatrix} \boldsymbol{K} & -\boldsymbol{B} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{B}^{\mathrm{T}} & \boldsymbol{0} & -\boldsymbol{R}_b & \boldsymbol{L}_b \\ \boldsymbol{0} & -\boldsymbol{R}_b^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_b^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{d} \\ \boldsymbol{\lambda}_b \\ \boldsymbol{\alpha} \\ \boldsymbol{u}_g \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f} \\ 0 \\ -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{f} \\ 0 \end{Bmatrix}, \qquad \boldsymbol{R}_b = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{R}\,, \qquad \boldsymbol{L}_b = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{L}\,. \tag{37}$$

Solving for the deformational displacement $\boldsymbol{d}$ from the first row of (37) gives

$$\boldsymbol{d} = \boldsymbol{F}(\boldsymbol{f} + \boldsymbol{B}\boldsymbol{\lambda}_b) \tag{38}$$

where $\boldsymbol{F}$ is the block-diagonal matrix of free–free flexibilities. Substituting (38) into the second row of (37) leads to a general flexibility equation:

$$\begin{bmatrix} \boldsymbol{F}_b & \boldsymbol{R}_b & -\boldsymbol{L}_b \\ \boldsymbol{R}_b^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \\ -\boldsymbol{L}_b^{\mathrm{T}} & \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\lambda}_b \\ \boldsymbol{\alpha} \\ \boldsymbol{u}_b \end{Bmatrix} = \begin{Bmatrix} -\boldsymbol{B}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{f} \\ -\boldsymbol{R}^{\mathrm{T}}\boldsymbol{f} \\ \boldsymbol{0} \end{Bmatrix}, \qquad \boldsymbol{F}_b = \boldsymbol{B}^{\mathrm{T}}\boldsymbol{F}\boldsymbol{B}\,. \tag{39}$$

Different choices of $\boldsymbol{B}$ lead to different flexibility methods. The DFM system (27) results if $\boldsymbol{B} = \boldsymbol{U}$. It is shown in [6] that selecting $\boldsymbol{B}$ as a null basis of $\boldsymbol{L}$ leads to a desirable choice for solving inverse problems. The CFM (1) can also be precipitated by a special choice of $\boldsymbol{B}$ along with variable transformations to a redundant basis.

## 9. Computing substructural RBMs and free–free flexibilities

The rigid body modes of a free–free substructure can be computed either by numerically extracting a null basis of the substructural stiffness matrix, or by considering its self-equilibrium. The former method has the advantage of generality and of being applicable to floating and fixed structures. Nonetheless, for free–free substructures with hundreds or thousands of elements, it has been found that extraction of a null basis is not only expensive but can lead to significant loss of accuracy. On that account we describe here the self-equilibrium method.

Suppose that free–free substructure $s$ contains $N_n^s$ nodes with the usual six degrees of freedom: three translations and three rotations, assigned at each. The translational nodal forces and nodal moments at node $n$, located at $(x_n,\ y_n,\ y_n)$, are denoted by $\boldsymbol{Q}_i^s = [Q_{xn}, Q_{yn}, Q_{zn}]^{\mathrm{T}}$ and $\boldsymbol{M}_i^s = [M_{xn}, M_{yn}, M_{zn}]^{\mathrm{T}}$, respectively. Self equilibrium requires that [17]:

$$\sum_{n=1}^{N_n^s} (\boldsymbol{R}_n^s)^{\mathrm{T}} \begin{Bmatrix} \boldsymbol{Q}_n^s \\ \boldsymbol{M}_n^s \end{Bmatrix} = \boldsymbol{0} \tag{40}$$

where

$$(\boldsymbol{R}_n^s)^{\mathrm{T}} = \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{0} \\ \boldsymbol{\chi}_n & \boldsymbol{I}_3 \end{bmatrix}, \qquad \boldsymbol{\chi}_n = \begin{bmatrix} 0 & -(z_n - z_0) & (y_n - y_0) \\ (z_n - z_0) & 0 & -(x_n - x_0) \\ -(y_n - y_0) & (x_n - x_0) & 0 \end{bmatrix} \tag{41}$$

where $(x_0, y_0, z_0)$ are the coordinates of a reference point, which for convenience may be taken to be equal to the average of the substructure node coordinates. The first three rows correspond to the translational rigid body modes, and the last three to three rotational rigid body modes. For substructures containing other node freedom configurations the procedure is similar. It follows that the (unnormalized) RBM for the substructure is obtained by stacking those nodal matrices:

$$\boldsymbol{R}^s = \begin{bmatrix} \boldsymbol{R}_1^s \\ \vdots \\ \boldsymbol{R}_{N_n^s}^s \end{bmatrix}. \tag{42}$$

This matrix may be then orthonormalized if convenient for subsequent computations. The RBM for the set of all substructures is the block diagonal matrix

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}^1 & & \cdots & \\ & \boldsymbol{R}^2 & \cdots & \\ & & \ddots & \boldsymbol{R}^{N^s} \end{bmatrix}. \tag{43}$$

We now turn to the computation of the free–free flexibility for substructure $s$ and assume that $\boldsymbol{K}^s$ has been assembled by the DSM executed for that substructure. Note that

$$\boldsymbol{u}^s = \boldsymbol{d}^s + \boldsymbol{R}^s\boldsymbol{\alpha}^s\,. \tag{34}$$

This is further partitioned as

$$\begin{Bmatrix} \boldsymbol{u}_c^s \\ \boldsymbol{u}_f^s \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{d}_c^s \\ \boldsymbol{d}_f^s \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{R}_c^s \boldsymbol{\alpha}^s \\ \boldsymbol{R}_f \boldsymbol{\alpha}^s \end{Bmatrix} \tag{44}$$

where $\boldsymbol{R}_c^s$ is a square invertible submatrix. Solving for $\boldsymbol{\alpha}^s$ from the first row of (44), we obtain $\boldsymbol{\alpha}^s = (\boldsymbol{R}_c^s)^{-1}(\boldsymbol{u}_c^s - \boldsymbol{d}_c^s)$. We now introduce a deformation measure that represents a relative deformation of $f$-freedoms with respect to the $c$-freedoms:

$$\boldsymbol{v}^s = \boldsymbol{d}_f^s - \boldsymbol{R}_f^s(\boldsymbol{R}_c^s)^{-1}\boldsymbol{d}_c^s = \boldsymbol{T}\boldsymbol{u}^s, \qquad \boldsymbol{T} = [-\boldsymbol{H} \quad \boldsymbol{I}], \qquad \boldsymbol{H} = \boldsymbol{R}_f^s(\boldsymbol{R}_c^s)^{-1} \tag{45}$$

where the $s$ superscript in $\boldsymbol{T}$ and $\boldsymbol{H}$, as well as in $\boldsymbol{C}$ and $\boldsymbol{K}_v$ below, is suppressed for brevity. Using this relation, $\boldsymbol{K}^s$ can be shown to be

$$\boldsymbol{K}^s = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{K}_v\boldsymbol{T}, \tag{46}$$

where $\boldsymbol{K}_v$ is obtained by eliminating the rows and columns of $\boldsymbol{K}^s$ that correspond to the degrees of freedom of $\boldsymbol{u}_c^s$ in (44). Therefore, the free–free flexibility $\boldsymbol{K}^s$ can be obtained from

$$\boldsymbol{F}^s = \boldsymbol{P}_r^s[\boldsymbol{K}^s + \boldsymbol{R}^s(\boldsymbol{R}^s)^{\mathrm{T}}]^{-1} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{K}_v^{-1}\boldsymbol{C}\boldsymbol{T}, \qquad \boldsymbol{C} = \boldsymbol{I} - \boldsymbol{H}[\boldsymbol{I} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}]^{-1}\boldsymbol{H}^{\mathrm{T}}. \tag{47}$$

It should be noted that for substructures with many DOFs, $\boldsymbol{K}_v$ is sparse while the symmetric matrix $[\boldsymbol{I} + \boldsymbol{H}^{\mathrm{T}}\boldsymbol{H}]$ appearing in $\boldsymbol{C}$ is at most a $(6 \times 6)$ matrix. Taking advantages of these properties is important to make the computation of $\boldsymbol{F}^s$ efficient.

There is an interesting relationship between these techniques and the computation of the condensed stiffness matrix in terms of the boundary freedoms, an operation common in finite element analysis. Here, the conventional method partitions the substructural stiffness matrix as

$$\boldsymbol{K}^s = \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{K}_{ib} \\ \boldsymbol{K}_{bi} & \boldsymbol{K}_{bb} \end{bmatrix} \tag{48}$$

from which the condensed or Schur-complement stiffness matrix follows as $\boldsymbol{K}_b = \boldsymbol{K}_{bb} - \boldsymbol{K}_{bi}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{ib}$. When the number of the internal freedoms far exceeds that of the boundary freedoms, the factorization of $\boldsymbol{K}_{ii}$ can be expensive. In that case a more economical method is to compute $\boldsymbol{K}_{bb}^c$ using the boundary-node flexibility $\boldsymbol{F}_b^s$ by the following formula:

$$\boldsymbol{K}_b = \boldsymbol{P}_r^s[\boldsymbol{P}_r^s\boldsymbol{F}_b^s\boldsymbol{P}_b^s + \boldsymbol{R}_b^s[(\boldsymbol{R}_b^s)^{\mathrm{T}}\boldsymbol{R}_b^s]^{-1}(\boldsymbol{R}_b^s)^{\mathrm{T}}]^{-1} \tag{49}$$

where $\boldsymbol{R}_b^s$ is $\boldsymbol{R}^s$ evaluated at the boundary nodes. This and related matrices are used for preconditioning of the projected residuals for iterative solutions of several methods presented in [4].

## 10. Illustrative examples

### 10.1. Free–free truss

Before showing a continuum example, we demonstrate the present DFM using one simple example: a five-DOF free–free spring shown in Fig. 6.

The elemental and global stiffness matrices, $\boldsymbol{K}$ and $\boldsymbol{K}_g$, respectively, are given by

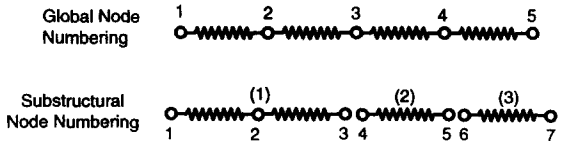


Fig. 6. Partition of five-DOF free–free spring system into three substructures.

$$
\boldsymbol{K}=\begin{bmatrix}
1 & -1 & & & & & & \\
-1 & 11 & -10 & & & & & \\
 & -10 & 10 & & & & & \\
 & & & 100 & -100 & & & \\
 & & & -100 & 100 & & & \\
 & & & & & 1000 & -1000 & \\
 & & & & & -10000 & 1000
\end{bmatrix},
$$

$$
\boldsymbol{K}_g=\begin{bmatrix}
1 & -1 & & & \\
-1 & 11 & -10 & & \\
 & -10 & 110 & -100 & \\
 & & -100 & 1100 & -1000 \\
 & & & -1000 & 1000
\end{bmatrix} \tag{50}
$$

where omitted entries are zero. The element assembly operator $\boldsymbol{L}$, the displacement compatibility matrix $\boldsymbol{U}$, and the rigid-body modes $\boldsymbol{R}$ are obtained as

$$
\boldsymbol{L}=\begin{bmatrix}
1 & & & & & \\
 & 1 & & & & \\
 & & 1 & & & \\
 & & 1 & & & \\
 & & & 1 & & \\
 & & & 1 & & \\
 & & & & 1
\end{bmatrix},\quad
\boldsymbol{U}=\begin{bmatrix}
1 & 0 & & & & \\
0 & 0 & & & & \\
0 & 1 & & & & \\
 & & 1 & 0 & & \\
 & & 0 & 1 & & \\
 & & & & 1 & 0 \\
 & & & & 0 & 1
\end{bmatrix},\quad
\boldsymbol{R}=\begin{bmatrix}
1 & \\
1 & \\
1 & \\
 & 1 \\
 & 1 \\
 & \\
 & 1
\end{bmatrix}. \tag{51}
$$

With (50) and (51), all the necessary quantities needed for the DFM equation (38) are now given. In practice, for computational convenience the rigid-body modes are usually normalized.

### *10.2. Cantilever plate model*

As a continuum example, the square cantilever plate used as benchmark model problem by Farhat and Mandel [35] is analyzed. The plate is discretized with the 3-node ANDES plate bending elements of Militello and Felippa [36] and subjected to a uniform lateral load. The deformed shape of the plate for the case of a 16-by-16 elements mesh is shown in Fig. 7.

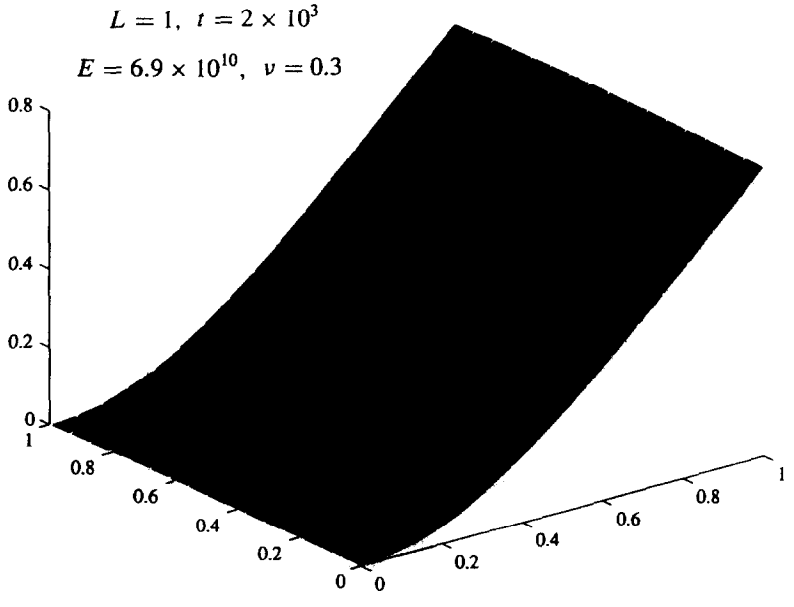


Fig. 7. Deformed shape of $16 \times 16$ mesh for square cantilever plate.

Table 2
Scalability test of FETI-1, FETI-2 and AFETI for cantilever plate problem (stopping criterion: global relative 2-norm residual $\leq 10^{-6}$)

| $h/l$ | Number of subdomains | Number of iterations FETI-1 | Number of iterations A-FETI | Number of iterations FETI-2 | Lagrange multipliers A-FETI |
|---|---|---|---|---|---|
| 1/8 | 4 | | 1 | | 102 |
| 1/8 | 16 | | 19 | | 306 |
| 1/16 | 4 | | 14 | | 198 |
| 1/16 | 16 | | 29 | | 594 |
| 1/32 | 16 | | 40 | | 1170 |
| 1/40 | 16 | 69 | 41 | 34 | 1396 |
| 1/80 | 16 | 82 | 45 | 41 | 2898 |
| 1/90 | 36 | 154 | 81 | 43 | 5430 |
| 1/120 | 64 | 238 | 120 | 50 | 10122 |

Table 2 presents conjugate-gradient iteration counts when this problem is solved using three different versions of the FETI parallel solvers. FETI-1 is the original, differentially partitioned FETI solver developed by Farhat and coworkers [28]. FETI-2 incorporates a more refined coarse mesh solver which significantly cuts the number of iterations for plates and shell problems [35]. A-FETI is the algebraically partitioned FETI based on projected DFM equation (26) and the preconditioner (27); implementation details are presented in [4]. The ratio $h/l$ designates the element mesh size vs. the substructure size.

As can be observed the number of A-FETI iterations falls in between those of FETI-1 and FETI-2, and are closer to the latter. FETI-2 has been shown to be scalable in the sense that the number of iterations should grow only as a multiple of $\log(l/h)$ under some coercivity conditions. That scalability has been recently tested on realistic aerospace problems with up to 4 million equations [37]. Relative CPU costs have not been compared because the implementations are not yet comparable on a maturity basis.

## 11. Concluding remarks

The DFM uses the same model information as the Direct Stiffness Method and does not require user inputs on redundant selection. The substructure flexibility matrices can be obtained from the stiffness equations and geometrically constructed rigid body modes. Consequently, the DFM can be implemented, as an alternative solution algorithm, within the architecture of a standard FEM code with substructuring capabilities.

Based on our experience, the DFM appears to be attractive for the following special applications.

(1) As a variant of the original FETI method [28] for scalable solution of large systems on massively parallel computers. This version was originally derived through algebraic partitioning of the DSM equations [3,4]. Preliminary experience to date has been encouraging.

(2) The inverse problem of extracting substructural flexibility matrices from the global flexibility constructed from experimental measurements. An important application of this procedure is damage localization [1]. A related problem is that of structural optimization under internal force constraints.

(3) The treatment of structures with rigid members or inclusions. This can be done simply by setting the appropriate compliances or flexibilities to zero, without incurring numerical difficulties. A promising application is rigid multibody dynamics with selective flexible members. Note that for this case the DFM behaves as dual of the DSM. In the latter, holes or voids are easily handled by setting stiffnesses to zero.

(4) The use of underintegrated isoparametric elements can be made attractive by treating the spurious modes as an extension of the RBM basis. This is being exploited in the development of nonlinear 3D analysis using one-point integrated elements while avoiding problem-dependent hourglass control strategies [38].

A yet unsettled part of the DFM is the best way to apply nodal support conditions, as well as the effective treatment of multipoint kinematic constraints such as rigid links and incompressibility. A algebraic treatment in Step 3 for the former was described. Although conceptually straightforward, this approach has two drawbacks:

(1) The handling of *partly supported* floating substructures is numerically fuzzy because numerical rank detection is inherently a singular perturbation problem.

(2) Program modularity is hindered by passing boundary condition information to substructure processors.

Ideally, the application of all such conditions should 'wait for the last moment' as in the DSM. This would allow, for example, more efficient handling of multiple load cases as well as variable-stiffness support conditions.

The last drawback may be circumvented by a *penalty method* presently under investigation, in which all single-freedom support boundary conditions are applied through fictitious linear or torsional spring elements. For infinitely-rigid springs it is sufficient to carry along the support reactions as interaction forces. If successful, this approach would allow all substructures *to be treated as free–free* up to Step 4, hence eliminating the modularity problem for single-freedom constraints. A key difference as regard penalty elements in the DSM should be noted: their stiffness can be made exactly infinite by setting their compliance to zero, an operation which does not cause numerical ill-conditioning. The treatment of rigid links and other multifreedom constraints could be done in principle through similar techniques but more research on their practical implementation is needed.

## Acknowledgments

The present work has been supported by the National Science Foundation under NSF/HPCC Grant ASC-9217394 and by Sandia National Laboratories under Contracts AS-5666 and AS-9991.

## References

[1] K.C. Park and K.F. Alvin, Extraction of substructural flexibility from measured global modes and mode shapes, Proc. 1996 AIAA SDM Conference, Paper No. AIAA 96-1297, Salt Lake City, Utah, April 1996; submitted to AIAA J.

[2] W.S. Hwang, K.W. Belvin and K.C. Park, Design of complex vibration control systems based on spatial energy transmission patterns, Proc. 1995 AIAA SDM Conference, Paper No. AIAA 95-1381, April 18–21 1995, New Orleans, LA.; submitted to AIAA J. Guidance, Control Dynam.

[3] K.C. Park, M.R. Justino F. and C.A. Felippa, An algebraically partitioned FETI method for parallel structural analysis: algorithm description, Center for Aerospace Structures Report CU-CAS-96-06, University of Colorado, Boulder, CO, 1996; to appear in Int. J. Numer. Methods Engrg., 1997.

[4] M.R. Justino F., K.C. Park and C.A. Felippa, An algebraically partitioned FETI method for parallel structural analysis: performance evaluation, Center for Aerospace Structures Report CU-CAS-96-12, University of Colorado, Boulder, CO, 1996; to appear in Int. J. Numer. Methods Engrg., 1997.

[5] M.R. Justino F. and K.C. Park, A matrix-free algebraic FETI method for quasistatic nonlinear structural analysis, Center for Aerospace Structures, Report No. CU-CAS-96-19, University of Colordo, 1996; to be presented at the 4th U.S. National Congress on Computational Mechanics, 6–8 August 1977, San Francisco, CA.

[6] K.C. Park and C.A. Felippa, A variational framework for solution method developments in structural mechanics, Center for Aerospace Structures, Report No. CU-CAS-96-22, University of Colorado, 1966; submitted to J. Appl. Mech.

[7] K.C. Park, Partitioned transient analysis procedures for coupled-field problems: stability analysis, J. Appl. Mech. 47 (1980) 370–376.

[8] C.A. Felippa and K.C. Park, Staggered transient analysis procedures for coupled-field mechanical systems: formulation, Comput. Methods Appl. Mech. Engrg. 24 (1980) 61–111.

[9] K.C. Park and C.A. Felippa, Partitioned analysis of coupled systems, in: T. Belytschko and T.J.R. Hughes, eds., Computational Methods for Transient Analysis (North-Holland Publishing Co., 1983) 157–219.

[10] S. Levy, Computation of influence coefficients for aircraft structures with discontinuities and sweepback, J. Aero. Sci. 14 (1947) 547–560.

[11] T. Rand, An approximate method for computation of stresses in sweptback wings, J. Aero. Sci. 18 (1951) 61–63.

[12] B. Langefors, Analysis of elastic structures by matrix coefficients, with special regard to semimonocoque structures, J. Aero. Sci. 19 (1952) 451–458.

[13] L.B. Wehle and W. Lansing, A method for reducing the analysis of complex redundant structures to a routine procedure, J. Aero. Sci. 19 (1962) 677–684.

[14] J.H. Argyris and S. Kelsey, Energy Theorems and Structural Analysis (Butterworths, London, 1960); reprinted from Aircraft Engrg. 26 Oct–Nov 1954 and 27 April–May, 1955.

[15] P.H. Denke, A general digital computer analysis of statically indeterminate structures, NASA Tech. Note D-1366, 1962.

[16] J. Robinson, Structural Matrix Analysis for the Engineer (Wiley, New York, 1966).

[17] J.S. Przemieniecki, Theory of Matrix Structural Analysis (McGraw-Hill, 1968) (Dover edition 1986).

[18] R.H. MacNeal, The MacNeal Schwendler Corporation: The First Twenty Years (Gardner Litograph, Buena Park, CA, 1988).

[19] I. Kaneko, M. Lawo and G. Thierauf, On computational procedures for the Force Method, Int. J. Numer. Methods Engrg. 18 (1982) 1469–1495.

[20] M.W. Berry, M.T. Heath, I. Kaneko, M. Lawo, R.J. Plemmons and R.C. Ward, An algorithm to compute a sparse basis of the null space, Numer. Math. 47 (1985) 483–504.
[21] I. Kaneko and R.J. Plemmons, Minimum norm solutions to linear elastic analysis problems, Int. J. Numer. Methods Engrg. 20 (1984) 983–998.
[22] R.J. Plemmons and R.E. White, Substructuring methods for computing the nullspace of equilibrium matrices, SIAM J. Matrix Anal. Appl. 1 (1990) 1–22.
[23] S.N. Patnaik, An Integrated Force Method for discrete analysis, Int. J. Numer. Methods Engrg. 6 (1973) 237–251.
[24] S.N. Patnaik, The variational energy formulation for the Integrated Force Method, AIAA J. 24 (1986) 129–136.
[25] S.N. Patnaik and H. Satish, Analysis of continuum using the boundary compatibility conditions of integrated Force Method, Comput. Struct. 34 (1990) 287–295.
[26] C.A. Felippa, Will the Force Method come back?, J. Appl. Mech. 54 (1987) 728–729.
[27] C.A. Felippa, Parametric unification of matrix structural analysis: classical formulation and d-Connected Mixed Elements, Finite Elem. Anal. Des. 21 (1995) 45–74.
[28] C. Farhat and F.X. Roux, Implicit parallel processing in structural mechanics, Comput. Mech. Adv. 2(1) (1994) 1–124.
[29] R.H. Gallagher, Private communication to K.C. Park, 1997.
[30] E.C. Pestel and F.A. Leckie, Matrix Methods in Elastomechanics (McGraw-Hill, New York, 1963).
[31] M.J. Turner, R.W. Clough, H.C. Martin and L.J. Topp, Stiffness and deflection analysis of complex structures, J. Aeron. Sci. 23 (1956) 805–824.
[32] M.J. Turner, The direct stiffness method of structural analysis, Structural and Materials Panel Paper, AGARD Meeting, Aachen, Germany, 1959.
[33] M.J. Turner, H.C. Martin and R.C. Weikel, Further development and applications of the stiffness method, AGARD Structures and Materials Panel, Paris, France, July 1962, in: B.M. Fraeijs de Veubeke, eds., AGARDograph 72: Matrix Methods of Structural Analysis(1964) 203–266.
[34] C.A. Felippa, K.C. Park and M.R. Justino F., A free–free flexibility matrix for structural analysis, Center for Aerospace Structures Report, University of Colorado, Boulder, CO, in preparation.
[35] C. Farhat and J. Mandel, The two-level FETI method for static and dynamic plate problems—Part I: An optimal iterative solver for biharmonic systems, Center for Aerospace Structures, University of Colorado, Report Number CU-CAS-95-23, Boulder, CO, October 1995.
[36] C. Militello and C.A. Felippa, The first ANDES elements: 9-dof plate bending triangles, Comput. Methods Appl. Mech. Engrg. 93 (1991) 217–246.
[37] C. Farhat, Private communication, 1997.
[38] K.C. Park, The Direct Flexibility Method makes spurious-mode stabilization unnecessary for one-point integrated elements, Center for Aerospace Structures, University of Colorado, Report Number CU-CAS-97-02, Boulder, CO, January 1997.