ELSEVIER

Comput. Methods Appl. Mech. Engrg. 191 (2002) 2111–2140

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

The construction of free–free flexibility matrices for multilevel structural analysis

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

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

Received 27 August 2001; received in revised form 20 November 2001; accepted 23 November 2001

Abstract

This article considers a generalization of the classical structural flexibility matrix. It expands on previous papers by taking a deeper look at computational considerations at the substructure level. Direct or indirect computation of flexibilities as "influence coefficients" has traditionally required pre-removal of rigid body modes by imposing appropriate support conditions, mimicking experimental arrangements. With the method presented here the flexibility of an individual element or substructure is directly obtained as a particular generalized inverse of the free–free stiffness matrix. This generalized inverse preserves the stiffness spectrum. The definition is element independent and only involves access to the stiffness generated by a standard finite element program and the separate construction of an orthonormal rigid-body mode basis. The free–free flexibility has proven useful in special application areas of finite element structural analysis, notably massively parallel processing, model reduction and damage localization. It can be computed by solving sets of linear equations and does not require processing an eigenproblem or performing a singular value decomposition. If substructures contain thousands of d.o.f., exploitation of the stiffness sparseness is important. For that case this paper presents a computation procedure based on an exact penalty method, and a projected rank-regularized inverse stiffness with diagonal entries inserted by the sparse factorization process. These entries can be physically interpreted as penalty springs. This procedure takes advantage of the stiffness sparseness while forming the full free–free flexibility, or a boundary subset, and is backed by an in-depth null space analysis for robustness. © 2002 Published by Elsevier Science B.V.

Keywords: Free–free flexibility; Free–free stiffness; Finite elements; Generalized inverse; Modified inverse; Projectors; Structures; Substructures; Penalty springs; Parallel processing

1. Motivation

The direct or indirect computation of structural flexibilities as "influence coefficients" has traditionally required precluding rigid body modes (RBMs) by imposing appropriate support conditions. This analytical

* Corresponding author. Tel.: +1-303-492-6547; fax: +1-303-492-4990.
E-mail address: carlos@titan.colorado.edu (C.A. Felippa).

0045-7825/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0045-7825(01)00379-6

approach mimics time-honored experimental practices for static tests, in which forces are applied to a supported structure, and deflections measured. See Fig. 1 for a classical example.

There are several application areas, however, for which it is convenient to have an expression for the flexibility matrix of a free–free structure, substructure or element. Important applications include domain decomposition for FETI-type iterative parallel solvers [1–5] as well as system identification and damage localization [6–9]. The qualifier "free–free" is used here to denote that all rigid body motions are unrestrained. This entity will be called a *free–free flexibility* matrix, and denoted by **F**.

In the symmetric case, which is the most important one in FEM modeling, the free–free flexibility represents the true dual of the well-known free–free stiffness matrix **K**. More specifically, **F** and **K** are the pseudo-inverses (also called the Moore–Penrose generalized inverses) of each other. The general expression for the pseudo-inverse of a singular symmetric matrix such as **K** involves its singular value decomposition (SVD) or, equivalently, knowledge of the eigensystem of **K**; see, e.g., [10]. That kind of analysis is not only expensive, but notoriously sensitive to rank decisions when carried out in floating-point arithmetic. That is, when can a small singular value be replaced by zero? Such decisions depend on the problem and physical units as well as the working computer precision.

Explicit expressions are presented here that relate **K** and **F** but involve only matrix inversions or, equivalently, the solution of linear systems. These expressions assume the availability of a basis matrix **R** for the RBMs, which span the null space of **K**. Often **R** may be constructed by geometric arguments separately from **K** and **F**. Because no eigenanalysis is involved, the formulas are suitable for symbolic manipulation in the case of simple individual elements.

The free–free flexibility has been introduced in previous papers [11,12], but these have primarily focused on theoretical and "system level use" considerations. Four practical aspects are studied here in more detail at the individual substructure level:

1. Relations between the free–free boundary-reduced stiffness and flexibility matrices (Section 4).
2. Cleaning up "RBM-polluted" matrices by projector filtering (Section 5).
3. Use of congruential forms in symbolic processing of individual elements (Section 6).
4. Efficient computation of **F** for substructures of moderate or large size (Section 7). The procedure relies on a general expression that involves a regularization matrix and the RBM projector.

This paper is organized as follows. Section 2 gives terminology and mathematical preliminaries. Section 3 introduces the free–free flexibility and its properties. Section 4 discusses the reduction of these expressions to boundary freedoms. Section 5 discusses projector filtering to clean up RBM-polluted matrices. Section 6 derives the free–free flexibility of selected individual elements. Section 7 addresses the issue of efficient computation of **F**, given **K** and **R**, for arbitrary substructures. Section 8 summarizes our key findings. Appendix A collects background material on generalized inverses and the inversion of modified matrices.

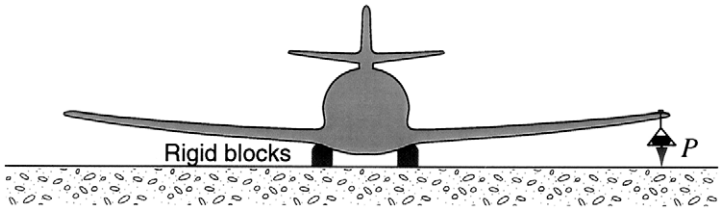

Fig. 1. Old-fashioned experimental determination of wing flexibilities. The aircraft is set on rigid blocks (not its landing gear) and a load P applied at the wing tip torque-center. The ratio of tip deflection to P gives the overall bending flexibility coefficient. The other key coefficient for flutter speed calculations is the torsional flexibility obtained by measuring the pitch rotations produced by an offset tip load.

2. Substructures terminology

A substructure is defined here as an assembly of finite elements that does not possess zero-energy modes aside from RBMs (see Fig. 2). Mathematically, the null space of the substructure stiffness contains only RBMs. (This restriction may be selectively relaxed, as further discussed in Section 7.9.) Substructures include individual elements and a complete structure as special cases. The total number of d.o.f. of the substructure is called N_F.

Should it be necessary, substructures are identified by a superscript enclosed in brackets; for example, $\mathbf{K}^{[3]}$ is the stiffness of substructure 3. Because this paper deals only with individual substructures, that superscript will be omitted to reduce clutter.

Fig. 3(a) depicts a 2D substructure, and the force systems acting on its nodes from external agents. The applied forces $\mathbf{f}_a$ are given as data. The interaction forces $\boldsymbol{\lambda}$ are exerted by other connected substructures. If the substructure is supported or partly supported, it is rendered *free–free* on replacing the supports by reaction forces $\mathbf{f}_s$ as shown in Fig. 3(b). The total force vector acting on the substructure is the superposition

$$\mathbf{f} = \mathbf{f}_a + \mathbf{f}_s + \boldsymbol{\lambda}, \tag{1}$$

where each vector has length N_F. Vectors $\boldsymbol{\lambda}$ and $\mathbf{f}_s$ are completed with zero entries as appropriate for conformity.

At each node considered as a free body, the internal force $\mathbf{p}$ is defined to be the resultant of the acting forces, as depicted in Fig. 3(c). Hence the statement of node by node equilibrium is

$$\mathbf{f} - \mathbf{p} = \mathbf{0} \quad \text{or} \quad \mathbf{f}_a + \mathbf{f}_s + \boldsymbol{\lambda} - \mathbf{p} = \mathbf{0}. \tag{2}$$

For some applications, notably iterative solvers, it is natural to view support reactions as interaction forces by regarding the "ground" as another substructure (often identified by a zero number). In such case vector $\mathbf{f}_s$ is merged with $\boldsymbol{\lambda}$.

As for the kinematics, the free–free substructure has $N_R > 0$, linearly independent RBMs. Rigid motions are characterized through the RBM-basis matrix $\mathbf{R}$, dimensioned $N_F \times N_R$, such that any rigid node displacement can be represented as $\mathbf{u}_R = \mathbf{R}\mathbf{a}$, where $\mathbf{a}$ is a vector of N_R modal amplitudes. The total displacement vector $\mathbf{u}$ can be written as a superposition of deformational and rigid motions:

$$\mathbf{u} = \mathbf{d} + \mathbf{u}_R = \mathbf{d} + \mathbf{R}\mathbf{a}, \tag{3}$$

Matrix $\mathbf{R}$ may be constructed by taking as columns N_R linearly independent rigid displacement fields evaluated at the nodes. For convenience those columns are assumed to be *orthonormalized* so that $\mathbf{R}^T\mathbf{R} = \mathbf{I}_R$, the identity matrix of order N_R. The orthogonal projector associated with $\mathbf{R}$ is the symmetric idempotent matrix

$$\mathbf{P} = \mathbf{I} - \mathbf{R}(\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T = \mathbf{I} - \mathbf{R}\mathbf{R}^T, \tag{4}$$

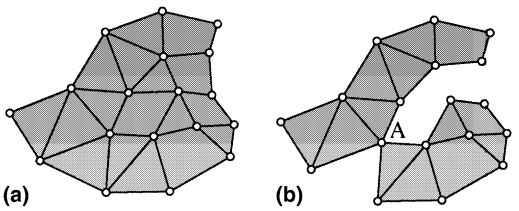

Fig. 2. Whereas (a) depicts a legal 2D substructure, the assembly (b) contains a nonRBM zero energy mode (relative rotation about point *A*). If such mechanisms are disallowed, (b) is not a legal substructure.

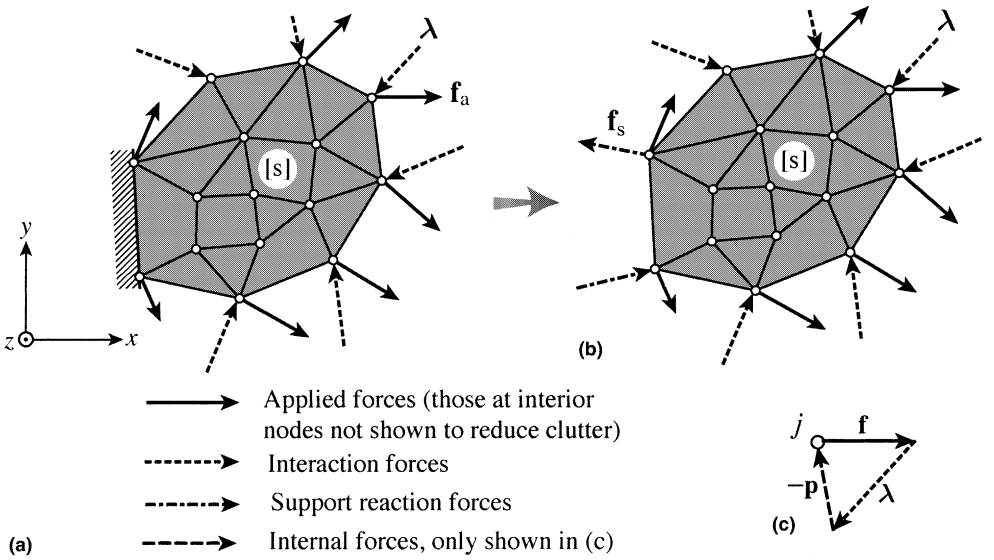

Fig. 3. A generic substructure [*s*] and the force systems acting on it. The top figures illustrate the conversion from a partly or fully supported configuration (a) to a free–free configuration (b) by replacing supports by reaction forces. Diagram (c) illustrates the self-equilibrium of a free-body node *j*.

where **I** (without subscript) is the $N_F \times N_F$ identity matrix. The last simplification in (4) arises from the assumed orthonormality of **R**. Note that $\mathbf{P}^2 = \mathbf{P}$ and $\mathbf{PR} = \mathbf{R}^T\mathbf{P} = \mathbf{0}$.

Application of the projector (4) to **u** extracts the deformational node displacements $\mathbf{d} = \mathbf{Pu}$. Subtracting these from the total displacements yields the rigid motions $\mathbf{u}_R = \mathbf{u} - \mathbf{d} = (\mathbf{I} - \mathbf{P})\mathbf{u} = \mathbf{R}\mathbf{R}^T\mathbf{u}$, which shows that $\mathbf{a} = \mathbf{R}^T\mathbf{u}$. Using the idempotent property $\mathbf{P}^2 = \mathbf{P}$ it is easy to verify that $\mathbf{d}^T\mathbf{u}_R = 0$. Fig. 4 gives a geometric interpretation of the decomposition (3).

Invoking the Principle of Virtual Work for the substructure under virtual rigid motions $\delta\mathbf{u}_R = \mathbf{R}\delta\mathbf{a}$ yields

$$\mathbf{R}^T\mathbf{f} = \mathbf{R}^T(\mathbf{f}_a + \mathbf{f}_s + \boldsymbol{\lambda}) = \mathbf{0} \tag{5}$$

as the statement of overall self-equilibrium for the substructure.

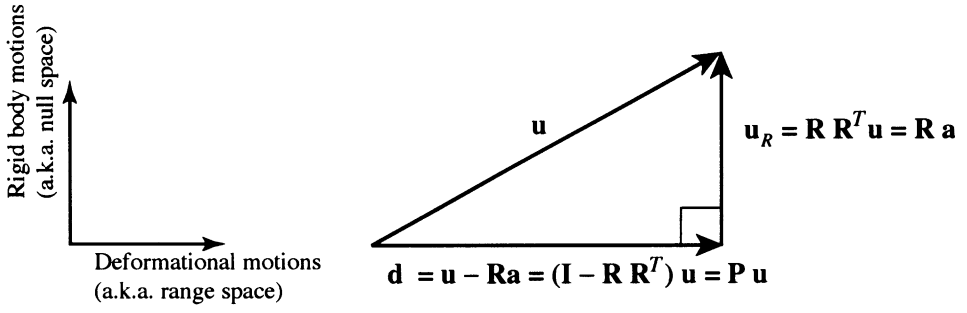

Fig. 4. Geometric interpretation of the projector **P**.

3. Stiffness and flexibility matrices

3.1. Definitions

The case of real-symmetric stiffness and flexibility matrices is the most important one in practice. This article only considers that case. The real-unsymmetric case, which arises in nonconservative and corotational FEM structural models, is briefly discussed in [11].

The well-known free–free stiffness matrix $\mathbf{K}$ of a linearly elastic substructure relates node displacements to node forces through the stiffness equations:

$$\mathbf{K}\mathbf{u} = \mathbf{f} = \mathbf{f}_a + \mathbf{f}_s + \boldsymbol{\lambda}, \quad \mathbf{K} = \mathbf{K}^T. \tag{6}$$

We assume throughout that $\mathbf{K}$ is *nonnegative*. If $\mathbf{K}$ models all rigid motions exactly, $\mathbf{R}^T\mathbf{K} = \mathbf{K}\mathbf{R}$ must vanish identically on account of (5). Such a stiffness matrix will be called *clean* as regards rigid body motions or, briefly, *RBM-clean*.

The free–free flexibility $\mathbf{F}$ of the substructure is defined by the expressions

$$\mathbf{F} = \mathbf{P}(\mathbf{K} + \mathbf{R}\mathbf{R}^T)^{-1} = (\mathbf{K} + \mathbf{R}\mathbf{R}^T)^{-1}\mathbf{P} = \mathbf{F}^T. \tag{7}$$

The symmetry of $\mathbf{F}$ follows from the spectral properties discussed in the following subsection. Using $\mathbf{F}$ we can write the free–free flexibility matrix equation dual to (6) as

$$\mathbf{F}\mathbf{f} = \mathbf{d} = \mathbf{P}\mathbf{u} = \mathbf{u} - \mathbf{u}_R. \tag{8}$$

Pre-multiplication by $\mathbf{R}^T$ and use of (5) shows that $\mathbf{F}\mathbf{R} = \mathbf{0}$.

If the substructure is fixed (that is, fully restrained against all rigid motions), matrix $\mathbf{R}$ is void. The definition (7) then collapses to that of the ordinary structural flexibility $\mathbf{F} = \mathbf{K}^{-1}$ whereas (8) reduces to $\mathbf{F}\mathbf{f} = \mathbf{u} - \mathbf{u}_R = \mathbf{u}$. Because of coalescence in the fully supported case, the same matrix symbol $\mathbf{F}$ can be used without risk of confusion.

3.2. Spectral and duality properties

The basic properties can be expressed in spectral language as follows. The free–free stiffness $\mathbf{K}$ has two kind of eigenvalues:

1. N_R *zero eigenvalues pertaining to rigid motions*. The associated null eigenspace is spanned by the columns of $\mathbf{R}$ because that basis matrix is assumed to be orthonormal.
2. $N_d = N_F - N_R$ *nonzero eigenvalues* λ_i. The associated orthonormalized "deformational eigenvectors" $\mathbf{v}_i$, which span the range space of $\mathbf{K}$, satisfy $\mathbf{K}\mathbf{v}_i = \lambda_i\mathbf{v}_i$, $\mathbf{R}^T\mathbf{v}_i = \mathbf{0}$ and $\mathbf{v}_i^T\mathbf{v}_j = \delta_{ij}$.

The eigenvectors of $\mathbf{K} + \mathbf{R}\mathbf{R}^T$ are identical to those of $\mathbf{K}$ but the RBM eigenvalues are raised to unity, giving the spectral decompositions

$$\mathbf{K} + \mathbf{R}\mathbf{R}^T = \sum_{i=1}^{N_d} \lambda_i\mathbf{v}_i\mathbf{v}_i^T + \mathbf{R}\mathbf{R}^T, \quad (\mathbf{K} + \mathbf{R}\mathbf{R}^T)^{-1} = \sum_{i=1}^{N_d} \frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^T + \mathbf{R}\mathbf{R}^T. \tag{9}$$

By construction the projector $\mathbf{P} = \mathbf{I} - \mathbf{R}\mathbf{R}^T$ has N_d unit eigenvalues whose range eigenspace is spanned by the $\mathbf{v}_i$, and the same null eigenspace as $\mathbf{K}$. Consequently, use of orthogonality properties yields the spectral decompositions

$$\mathbf{P} = \sum_{i=1}^{N_d} \mathbf{v}_i\mathbf{v}_i^T, \quad \mathbf{F} = \mathbf{P}(\mathbf{K} + \mathbf{R}\mathbf{R}^T)^{-1} = \sum_{i=1}^{N_d} \frac{1}{\lambda_i}\mathbf{v}_i\mathbf{v}_i^T. \tag{10}$$

The foregoing relations show that the three symmetric matrices $\mathbf{K}$, $\mathbf{F}$ and $\mathbf{P}$ have the same eigenvectors, and can be commuted at will. For example, $\mathbf{F}^{\alpha}\mathbf{K}^{\beta}\mathbf{P}^{\gamma} = \mathbf{K}^{\beta}\mathbf{P}^{\gamma}\mathbf{F}^{\alpha}$ for any integer exponents (α, β, γ). Commutativity proves the symmetrization in (7). Other useful relations that emanate from the spectral decompositions are

$$\mathbf{K} = \mathbf{P}(\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1} = (\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P} = \mathbf{K}^{\mathrm{T}}, \tag{11}$$

$$(\mathbf{K} + \alpha\mathbf{R}\mathbf{R}^{\mathrm{T}})(\mathbf{F} + \beta\mathbf{R}\mathbf{R}^{\mathrm{T}}) = \mathbf{I} + (\alpha\beta - 1)\mathbf{R}\mathbf{R}^{\mathrm{T}} = \mathbf{P} + \alpha\beta\mathbf{R}\mathbf{R}^{\mathrm{T}} \quad (\alpha,\ \beta \text{ arbitrary scalars}), \tag{12}$$

$$\mathbf{K}\mathbf{R} = \mathbf{F}\mathbf{R} = \mathbf{P}\mathbf{R} = \mathbf{0}, \tag{13}$$

$$\mathbf{K}\mathbf{F}\mathbf{K} = \mathbf{K}, \quad \mathbf{F}\mathbf{K}\mathbf{F} = \mathbf{F}, \quad (\mathbf{K}\mathbf{F})^{\mathrm{T}} = \mathbf{F}\mathbf{K} = \mathbf{K}\mathbf{F} = \mathbf{P}, \quad (\mathbf{F}\mathbf{K})^{\mathrm{T}} = \mathbf{K}\mathbf{F} = \mathbf{F}\mathbf{K} = \mathbf{P}, \tag{14}$$

$$\mathbf{R}^{\mathrm{T}}(\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{R} = \mathbf{I}, \quad \mathbf{R}^{\mathrm{T}}(\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{R} = \mathbf{I} \quad \text{(cf. Section A.3.2)}. \tag{15}$$

This relation catalog shows that $\mathbf{K}$ and $\mathbf{F}$ are *dual*, because exchanging them leaves all formulas intact.

Comparing (14) to the definitions in Section A.1 of the Appendix, it is seen that $\mathbf{F}$ and $\mathbf{K}$ are both pseudo-inverses and sg-inverses of each other: $\mathbf{F} = \mathbf{K}^{+} = \mathbf{K}^{\dagger}$ and $\mathbf{K} = \mathbf{F}^{+} = \mathbf{F}^{\dagger}$. The practical importance of (7) is that if $\mathbf{K}$ and $\mathbf{R}$ are given, $\mathbf{F}$ can be *computed by solving linear equations* without need of the more expensive eigenvalue analysis of $\mathbf{K}$. This is important for substructures containing hundreds or thousands of elements because linear equation solvers can take advantage of the natural sparsity of $\mathbf{K}$, as discussed in Section 7. The stiffness matrix can be efficiently generated by the direct stiffness method (DSM) using existing finite element libraries, whereas the construction of $\mathbf{R}$ from geometric arguments is straightforward as explained in Section 3.4.

3.3. Alternative expressions

The flexibility expressions (8) are actually the first two of the following 12 formulas for $\mathbf{F}$:

$$\begin{aligned}
&\mathbf{P}(\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},\\
&\mathbf{P}(\mathbf{P}\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{P}\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{P}\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},\\
&\mathbf{P}(\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},\\
&\mathbf{P}(\mathbf{P}\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{P}\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{P}\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}.
\end{aligned} \tag{16}$$

The seventh form, $\mathbf{P}(\mathbf{K}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}$, was found by Bott and Duffin [13] in a different context. Expressions (16) are equivalent in exact arithmetic if $\mathbf{K}$ is "RBM clean" in the sense that $\mathbf{K}\mathbf{R} = \mathbf{0}$. If $\mathbf{K}$ is, however, "polluted" so that $\mathbf{K}\mathbf{R} \neq \mathbf{0}$ the expressions (16) will generally yield different results for $\mathbf{F}$. Furthermore, matrices given by the formulas in the first three rows may not be symmetric. If $\mathbf{K}$ is polluted, the last three formulas are recommended, because the *filtered stiffness* $\overline{\mathbf{K}} = \mathbf{P}\mathbf{K}\mathbf{P}$ is guaranteed to be both symmetric and clean. This point is further examined in Section 5.

Similarly, if $\mathbf{F}$ is known, $\mathbf{K}$ may be computed from one of the 12 formulas

$$\begin{aligned}
&\mathbf{P}(\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},\\
&\mathbf{P}(\mathbf{P}\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{P}\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{P}\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},\\
&\mathbf{P}(\mathbf{F}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{F}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{F}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},\\
&\mathbf{P}(\mathbf{P}\mathbf{F}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}, \quad (\mathbf{P}\mathbf{F}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \mathbf{P}(\mathbf{P}\mathbf{F}\mathbf{P} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P},
\end{aligned} \tag{17}$$

which are equivalent if $\mathbf{F}\mathbf{R} = \mathbf{0}$.

Using the fact that $\mathbf{P}^k = \mathbf{P}$ for any integer $k > 0$, any $\mathbf{P}$ in (16) and (17) may be raised to arbitrary integer powers, but this generalization is vacuous.

3.4. Forming the RBM matrix

If the substructure is free–free and has no spurious modes, the construction of $\mathbf{R}$ by geometric arguments is straightforward. This is illustrated here for the 2D case depicted in Fig. 3. To facilitate satisfaction of orthogonality, it is convenient to place the x, y axes at the geometric mean of the N node locations of the substructure. Three independent RBMs are the x-translation $u_x = 1$, $u_y = 0$, the y-translation $u_x = 0$, $u_y = 1$ and the z-rotation $u_x = -y$, $u_y = x$. Evaluation at the nodes gives

$$\mathbf{R}^{\mathrm{T}} = \begin{bmatrix} 1 & 0 & 1 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 1 \\ -y_1 & x_1 & -y_2 & \cdots & x_N \end{bmatrix}. \tag{18}$$

The columns of this $\mathbf{R}$ are mutually orthogonal by construction. All that remains is to normalize them to unit length through division by $N^{1/2}$, $N^{1/2}$ and $[\sum_i (x_i^2 + y_i^2)]^{1/2}$, respectively. The 3D case is equally straightforward if the substructure has only translational freedoms. If rotational freedoms are present, explicit orthonormalization using Gram–Schmid or similar scheme may be required.

In geometrically nonlinear Lagrangian analysis a common occurrence is the loss of rotational RBMs dues to initial stress effects in the geometric stiffness, which results in a gain of rank of $\mathbf{K}$ with respect to the linear case. In plasticity analysis a loss of rank due to plastic flow mechanisms may occur. In either case the null space basis of $\mathbf{K}$ has to be appropriately adjusted.

4. Reduction to boundary freedoms

The expressions of $\mathbf{K}$, $\mathbf{F}$ and $\mathbf{R}$ used in the previous section pertain to all d.o.f. of the substructure. For many applications, notably the case of domain decomposition illustrated in Fig. 5, only the substructural boundary freedoms are involved. The interior d.o.f. are eliminated in some way. To illustrate the reduction scheme it is convenient to partition $\mathbf{F}$, $\mathbf{K}$ and $\mathbf{R}$ as follows:

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} \mathbf{F}_{bb} & \mathbf{F}_{bi} \\ \mathbf{F}_{ib} & \mathbf{F}_{ii} \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} \mathbf{R}_b \\ \mathbf{R}_i \end{bmatrix}. \tag{19}$$

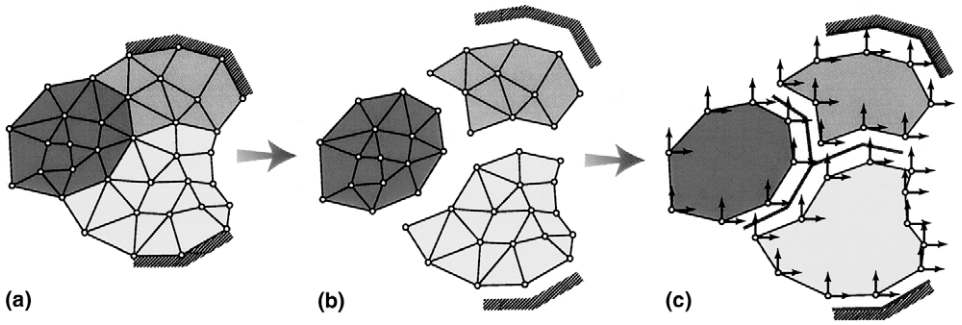

Fig. 5. Reduction to boundary freedoms in the domain decomposition of a FEM mesh. In solving the so-called coarse problem, which involves substructural interactions, only boundary node freedoms participate. The rightmost figure depicts such interactions occurring indirectly through intermediate connection frames, as in algebraic FETI methods [3,4].

In a free–free substructure, all interior d.o.f. are unconstrained (that is, nodal forces are known). Reduction to the boundary by static condensation produces the matrices

$$\mathbf{K}_b = \mathbf{K}_{bb} - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib}, \quad \mathbf{F}_b = \mathbf{F}_{bb}, \tag{20}$$

where $\mathbf{K}_b$ is the condensed stiffness matrix, also known as a Schur complement in the mathematical literature. Note that $\mathbf{K}_b$ and $\mathbf{F}_b$ are *not* dual: while the computation of $\mathbf{K}_b$ from $\mathbf{K}$ is involved, the reduction to a boundary flexibility $\mathbf{F}_b$ is trivial and can be simply done by extracting appropriate rows and columns of $\mathbf{F}$. Furthermore $\mathbf{K}_b$ is singular while $\mathbf{F}_b$ is usually nonsingular and well conditioned. These aspects are further addressed in Section 7, where efficient schemes to compute $\mathbf{F}_b$ from $\mathbf{K}$ and $\mathbf{R}$ are described.

Occasionally it is useful to pass directly from $\mathbf{F}_b$ to $\mathbf{K}_b$. For example, boundary flexibilities might be known from experimental data or a model reduction process. Denote the boundary projector by $\mathbf{P}_b = \mathbf{I} - \mathbf{R}_b\mathbf{S}\mathbf{R}_b^{\mathrm{T}}$, where $\mathbf{S} = (\mathbf{R}_b^{\mathrm{T}}\mathbf{R}_b)^{-1}$; note that since $\mathbf{R}_b$ is not generally orthonormal, the kernel scaling by $\mathbf{S}$ must be retained. Matrices $\mathbf{K}_b$ and $\overline{\mathbf{F}}_b = \mathbf{P}_b\mathbf{F}_b\mathbf{P}_b$ may be related by expressions (16) and (17), in which all matrices pertain to the boundary freedoms only. For example,

$$\mathbf{K}_b = \mathbf{P}_b\big(\overline{\mathbf{F}}_b + \mathbf{R}_b\mathbf{S}\mathbf{R}_b^{\mathrm{T}}\big)^{-1}, \quad \overline{\mathbf{F}}_b = \mathbf{P}_b\big(\mathbf{K}_b + \mathbf{R}_b\mathbf{S}\mathbf{R}_b^{\mathrm{T}}\big)^{-1}. \tag{21}$$

The dual of the first relation only returns the projection-filtered boundary flexibility $\overline{\mathbf{F}}_b = \mathbf{P}_b\mathbf{F}_b\mathbf{P}_b$, and not $\mathbf{F}_b$. Thus from a boundary stiffness matrix it is generally impossible to recover a full-rank boundary flexibility. This observation explains the superiority of $\mathbf{F}_b$ in model reduction processes.

5. Handling an RBM-polluted stiffness

If $\mathbf{K}\mathbf{R} \neq \mathbf{0}$ the stiffness matrix is said to be *polluted* as regard RBMs. Physically, application of a rigid motion to nodes produces nonzero forces. Assuming that all supports have been properly removed, RBM pollution may arise from one or more of the following sources:

(i) Inexact arithmetic in forming $\mathbf{K}$ or $\mathbf{R}$.
(ii) Kinematic defects in individual elements. For instance, some curved shell elements are known not to model rotational RBMs correctly.
(iii) Kinematic defects in assembly. The classical example is that of a smooth thin shell surface modeled by faceted elements without drilling d.o.f. Some FEM programs delete the surface-normal rotational freedom to preclude singularities in flat configurations, and in so doing pollute the assembled stiffness with respect to the other two rotational RBMs.

While (i) is typically benign, (ii) and (iii) can have serious effects. Unfortunately correction at the element and assembly level, respectively, may not be feasible because of "software legacy" conditions precluding changes. One way to "sanitize" $\mathbf{K}$ post-facto is to apply congruential filtering through the projector

$$\overline{\mathbf{K}} = \mathbf{P}^{\mathrm{T}}\mathbf{K}\mathbf{P} = \mathbf{P}\mathbf{K}\mathbf{P}. \tag{22}$$

The filtered matrix satisfies $\overline{\mathbf{K}}\mathbf{R} = \mathbf{0}$ since $\mathbf{P}\mathbf{R} = \mathbf{0}$ by construction.

A drawback of (22) at the substructure level is that $\overline{\mathbf{K}}$ is generally full. This jeopardizes computations where taking advantage of sparsity is important, such as those discussed in Section 7. Sparsity can be recovered by doing (22) at the individual element level before assembly. This approach, however, would not eliminate pollution from the assembly defects described under (iii) above.

The definition (7) of $\mathbf{F}$ is inconvenient if $\mathbf{K}$ is polluted because the spectral properties discussed in Section 3 are lost. And so is symmetry: $\mathbf{F} \neq \mathbf{F}^{\mathrm{T}}$. A symmetric $\mathbf{F}$ can be obtained in two ways. Either the stiffness is filtered before inversion or the filter applied after inversion:

$$\overline{\mathbf{F}} = (\overline{\mathbf{K}} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}, \quad \hat{\mathbf{F}} = \mathbf{P}(\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P}. \tag{23}$$

Both $\overline{\mathbf{F}}$ and $\hat{\mathbf{F}}$ are symmetric and RBM-clean. These two free–free flexibilities are generally different. As of this writing there is not enough experience to decide on which form is best, although $\hat{\mathbf{F}}$ is more computationally convenient if exploitation of stiffness sparseness is important.

6. Free–free flexibility of individual elements

The free–free flexibility of simple individual elements such as bars and beams may be computed directly from the definition (7) as illustrated in Examples 1 and 2. For 2D and 3D elements it is recommended to exploit the congruential forms outlined below.

6.1. Congruential forms

The following expressions follow from the pseudo-inverse forms listed in Appendix A.1.2:

$$\mathbf{K} = \mathbf{Q}^{\mathrm{T}}\mathbf{S}\mathbf{Q} = \mathbf{F}^{+}, \quad \mathbf{F} = \mathbf{Q}^{\mathrm{T}}\mathbf{C}\mathbf{Q} = \mathbf{K}^{+} \quad \text{if } \mathbf{Q}\mathbf{Q}^{\mathrm{T}} = \mathbf{I} \text{ and } \mathbf{S}\mathbf{C} = \mathbf{I}. \tag{24}$$

Here $\mathbf{Q}$ is an orthonormalized $N_{\mathrm{d}} \times N_{\mathrm{F}}$ strain–displacement matrix, $\mathbf{S}$ is a $N_{\mathrm{d}} \times N_{\mathrm{d}}$ deformational rigidity matrix, and $\mathbf{C} = \mathbf{S}^{-1}$ the corresponding compliance. Mathematically $\mathbf{P}_r^{\perp} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q}$ is the orthogonal projector complementary to $\mathbf{P}$.

The element stiffness matrix can be maneuvered into the form (24) through various algebraic or geometric transformations without need of an explicit spectral analysis. A particularly elegant scheme consists of orienting local axes along the principal directions of inertia of the element, as Section 6.4 illustrates.

For some nonsimplex continuum elements generalizations of (24) that may facilitate analytical inversion are

$$\mathbf{K} = \sum_{i}^{q} \mathbf{Q}_i^{\mathrm{T}}\mathbf{S}_i\mathbf{Q}_i, \quad \mathbf{F} = \sum_{i}^{q} \mathbf{Q}_i^{\mathrm{T}}\mathbf{C}_i\mathbf{Q}_i \quad \text{if } \mathbf{Q}_i\mathbf{Q}_j^{\mathrm{T}} = \mathbf{I} \text{ and } \mathbf{S}_i\mathbf{C}_i = \mathbf{I}. \tag{25}$$

Here the $\mathbf{Q}_i$ matrices span $q \geqslant 1$ mutually orthogonal subspaces, all of which are orthogonal to the RBM subspace. Elements that befit the template formulation [14], in which $q = 2$, may be maneuvered into form (25) through various orthogonalization techniques.

6.2. Example 1: Bar element

For the 1D 2-node bar element displayed in Fig. 6(a) direct application of the definition (8) yields

$$\begin{aligned} &\mathbf{K} = k\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}, \quad \mathbf{R} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{P} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{2k}\mathbf{K}, \\ &\mathbf{F} = \mathbf{P}(\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1} = \frac{1}{4k}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = \frac{1}{4k^2}\mathbf{K}. \end{aligned} \tag{26}$$

Here $k = EA/L$ is the axial (equivalent spring) stiffness. It is easily verified that the result $\mathbf{K} = 4k^2\mathbf{F}$ also holds for 2-node bars in two and three dimensions.

Alternatively one may exploit formula (24) by following the scheme:

$$\mathbf{K} = \mathbf{Q}^{\mathrm{T}}(2k)\mathbf{Q}, \quad \mathbf{Q} = \frac{1}{\sqrt{2}}[-1 \quad 1], \quad \mathbf{F} = \mathbf{Q}^{\mathrm{T}}\frac{1}{2k}\mathbf{Q} = \frac{1}{4k^2}\mathbf{Q}^{\mathrm{T}}(2k)\mathbf{Q} = \frac{1}{4k^2}\mathbf{K}, \tag{27}$$

which gives the same result.

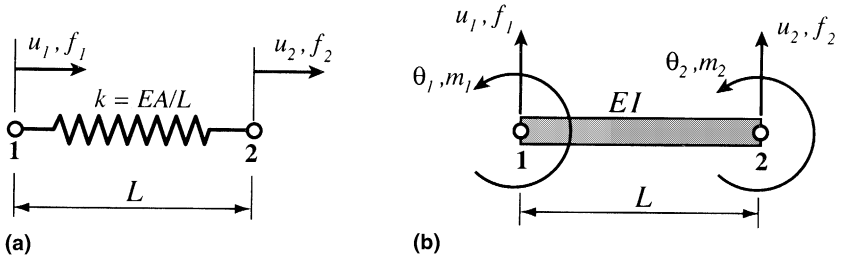

Fig. 6. Bar and beam elements for free–free flexibility examples.

6.3. Example 2: Plane beam element

For the 2-node, 4-d.o.f., Bernoulli–Euler prismatic plane beam element shown in Fig. 6(b),

$$\mathbf{K} = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}, \quad \mathbf{R} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & -L/\sqrt{4+L^2} \\ 0 & 2/\sqrt{4+L^2} \\ 1 & L/\sqrt{4+L^2} \\ 0 & 2/\sqrt{4+L^2} \end{bmatrix}, \tag{28}$$

whence

$$\mathbf{F} = \frac{L}{6EI(4+L^2)^2}\begin{bmatrix} 2L^2 & L^3 & -2L^2 & L^3 \\ L^3 & 24+12L^2+2L^4 & -L^3 & -24-12L^2-L^4 \\ -2L^2 & -L^3 & 2L^2 & -L^3 \\ L^3 & -24-12L^2-L^4 & -L^3 & 24+12L^2+2L^4 \end{bmatrix}. \tag{29}$$

Note that entries of **F** are not dimensionally homogeneous. This is a consequence of mixing translational and rotational nodal displacements and of normalizing the second column of **R**. A dimensionally homogeneous form can be obtained by scaling the rotational freedoms by a characteristic length, as done in the example of Fig. 11.

The approach based on the congruential form (25) starts from

$$\mathbf{Q}^{\mathrm{T}} = \frac{1}{\sqrt{2}}\begin{bmatrix} 0 & -1 & 0 & 1 \\ 2g/L & g & 2g/L & g \end{bmatrix}, \quad \mathbf{S} = \frac{EI}{L}\begin{bmatrix} 2 & 0 \\ 0 & \dfrac{6(4+L^2)}{L^2} \end{bmatrix}, \tag{30}$$

in which $g = 1/\sqrt{1+4/L^2}$. It is easy to verify that $\mathbf{Q}^{\mathrm{T}}\mathbf{S}^{-1}\mathbf{Q}$ reproduces (29).

6.4. Example 3: Plane stress 3-node triangle

The simplest continuum element is the 3-node, linear displacement, plane stress triangle illustrated in Fig. 7. The element has area A, uniform thickness h, and constant 3×3 elastic modulus matrix **E**. Axes x and y pass through the centroid 0. With the nodal displacements arranged as

$$\mathbf{u} = [\, u_{x1} \quad u_{y1} \quad u_{x2} \quad u_{y2} \quad u_{y3} \quad u_{y3} \,]^{\mathrm{T}}, \tag{31}$$

the free–free stiffness matrix has the well-known closed form expression

$$\mathbf{K} = hA\mathbf{B}^{\mathrm{T}}\mathbf{E}\mathbf{B}, \quad \mathbf{B} = \frac{1}{2A}\begin{bmatrix} y_{32} & 0 & y_{13} & 0 & y_{21} & 0 \\ 0 & x_{23} & 0 & x_{31} & 0 & x_{12} \\ x_{23} & y_{32} & x_{31} & y_{13} & x_{12} & y_{21} \end{bmatrix}, \tag{32}$$

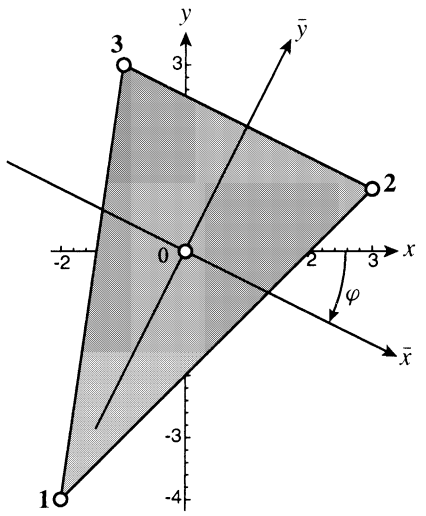

Fig. 7. 3-node plane stress triangular element. Displayed triangle has corners placed at (x,y) locations $(-2,-4)$, $(3,1)$ and $(-1,3)$. Area $A = 9/2$; principal direction angle $\varphi = -26.5°$.

in which $x_{ij} = x_i - x_j$ and $y_{ij} = y_i - y_j$. To develop the free–free flexibility, it is convenient to select axes $(\bar{x},\bar{y})$ with respect to which $\bar{\mathbf{B}}^{\mathrm{T}}\bar{\mathbf{B}}$ becomes a diagonal matrix, to allow use of (24). This is done by taking $(\bar{x},\bar{y})$ along the principal directions of inertia of the triangle. These are defined by the angle φ shown in Fig. 7. The calculations proceed as follows. Defining the nodal-lumped inertias

$$I_{xx} = y_{21}^2 + y_{32}^2 + y_{31}^2, \quad I_{yy} = x_{12}^2 + x_{23}^2 + x_{13}^2, \quad I_{xy} = x_{12}y_{21} + x_{23}y_{32} + x_{13}y_{31}. \tag{33}$$

φ and related trigonometric functions are obtained as

$$\begin{aligned} &\tan 2\varphi = \frac{I_{xy}}{I_{yy} - I_{xx}}, \quad \cos 2\varphi = \frac{1}{\sqrt{1+\tan^2 2\varphi}}, \quad \sin 2\varphi = \frac{\tan 2\varphi}{\sqrt{1+\tan^2 2\varphi}}, \\ &\cos^2\varphi = \tfrac{1}{2}(1+\cos 2\varphi), \quad \sin^2\varphi = \tfrac{1}{2}(1-\cos 2\varphi), \end{aligned} \tag{34}$$

in which the angle $-45° \leqslant \varphi \leqslant 45°$ is chosen; if $I_{xx} = I_{yy}$ one conventionally takes $\varphi = 0$. The principal inertias are

$$\bar{I}_{xx} = \tfrac{1}{2}(I_{xx} + I_{yy}) + R, \quad \bar{I}_{yy} = \tfrac{1}{2}(I_{xx} + I_{yy}) - R, \quad R = \sqrt{\frac{1}{4}(I_{yy} - I_{xx}) + I_{xy}^2}. \tag{35}$$

Introduce the strain transformation matrix and its inverse:

$$\mathbf{T}_e = \begin{bmatrix} \cos^2\varphi & \sin^2\varphi & \frac{1}{2}\sin 2\varphi \\ \sin^2\varphi & \cos^2\varphi & -\frac{1}{2}\sin 2\varphi \\ -\sin 2\varphi & \sin 2\varphi & \cos 2\varphi \end{bmatrix}, \quad \mathbf{T}_\sigma = \mathbf{T}_e^{-1} = \begin{bmatrix} \cos^2\varphi & \sin^2\varphi & -\frac{1}{2}\sin 2\varphi \\ \sin^2\varphi & \cos^2\varphi & \frac{1}{2}\sin 2\varphi \\ \sin 2\varphi & -\sin 2\varphi & \cos 2\varphi \end{bmatrix}. \tag{36}$$

Then define

$$\mathbf{J} = \frac{1}{4A^2}\begin{bmatrix} \bar{I}_{xx} & 0 & 0 \\ 0 & \bar{I}_{yy} & 0 \\ 0 & 0 & \bar{I}_{xx} + \bar{I}_{yy} \end{bmatrix}, \quad \mathbf{C} = \mathbf{T}_\sigma^{\mathrm{T}}\mathbf{J}\mathbf{T}_\sigma^{\mathrm{T}}\mathbf{E}^{-1}\mathbf{T}_\sigma\mathbf{J}\mathbf{T}_\sigma. \tag{37}$$

Here $\mathbf{C}$ is a modified constitutive matrix with dimensions of compliance (because $\mathbf{J}$ and $\mathbf{T}_\sigma$ are dimensionless). Using the formula (25) it may be verified that

$$\mathbf{F} = \frac{1}{Ah}\mathbf{B}^{\mathrm{T}}\mathbf{C}\mathbf{B}. \tag{38}$$

Because the effort involved in computing $\mathbf{C}$ is minor, the work involved in forming $\mathbf{F}$ is not too different from that required for the free–free stiffness (32). An alternative is to use the relation $\mathbf{F} = (Ah\mathbf{B}^{\mathrm{T}}\mathbf{E}\mathbf{B})^{+} = (Ah)^{-1}\mathbf{B}^{+}\mathbf{E}^{-1}(\mathbf{B}^{+})^{\mathrm{T}}$, which follows from Eqs. (A.4) and (A.5), with $\mathbf{B}^{+} = (\mathbf{B}^{\mathrm{T}}\mathbf{B})^{-1}\mathbf{B}$. This is easier to implement but lacks physical meaning.

7. Free–free flexibility of multielement substructures

We now pass to the case of a substructure that contains multiple elements. These may be of different types; for instance combinations of beams, plates and shells. The approach of forming $\mathbf{F}^{(e)}$ of each individual element and merging is unfeasible because there is no simple flexibility assembly procedure comparable to the DSM. It is better to assemble first the free–free substructure stiffness matrix $\mathbf{K}$ by the DSM. The geometric construction of the rigid body matrix $\mathbf{R}$ is also straightforward. From $\mathbf{K}$ and $\mathbf{R}$ one obtains $\mathbf{F}$, the computations being generally carried out in floating-point arithmetic. We review here approaches appropriate to small and large size substructures; the crossover on present computers being roughly 300 to 500 freedoms.

7.1. Direct calculation from definition

The simplest calculation of the free–free flexibility $\mathbf{F}$ is by solving either of the linear systems

$$(\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})\mathbf{F} = \mathbf{P} \quad \text{or} \quad (\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}})(\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}}) = \mathbf{I}, \tag{39}$$

for $\mathbf{F}$ or $\mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}}$, respectively, following factorization of the positive-definite symmetric matrix $\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}}$. In the second form $\mathbf{R}\mathbf{R}^{\mathrm{T}}$ is subtracted to get $\mathbf{F}$. Matrix $\mathbf{K} + \mathbf{R}\mathbf{R}^{\mathrm{T}}$ is dense since $\mathbf{R}\mathbf{R}^{\mathrm{T}}$ generally will "fill out" the matrix. Thus any sparseness initially present in $\mathbf{K}$ is lost. For substructures containing less than roughly $N_{\mathrm{F}} = 300$ d.o.f. this loss of sparsity is of little significance because the inversion of a full 300×300 positive-definite symmetric matrix requires on the order of 10^7 operations, which is insignificant on most computers (even PCs) endowed with floating-point hardware. The two systems (39) are equivalent if $\mathbf{K}$ is "RBM-clean" in the sense discussed in Section 5. If $\mathbf{K}$ is polluted the first form is preferable since $\mathbf{P}$ acts as a filter.

Equivalent to (39) are the block-bordered symmetric systems

$$\begin{bmatrix} \mathbf{K} & \mathbf{R} \\ \mathbf{R}^{\mathrm{T}} & -\mathbf{I}_{\mathrm{R}} \end{bmatrix} \begin{bmatrix} \mathbf{F} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{P} \\ \mathbf{0} \end{bmatrix}, \quad \begin{bmatrix} \mathbf{K} & \mathbf{R} \\ \mathbf{R}^{\mathrm{T}} & -\mathbf{I}_{\mathrm{R}} \end{bmatrix} \begin{bmatrix} \mathbf{F} + \mathbf{R}\mathbf{R}^{\mathrm{T}} \\ \tilde{\mathbf{G}} \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \tag{40}$$

in which $\mathbf{I}_{\mathrm{R}}$ denotes the $N_{\mathrm{R}} \times N_{\mathrm{R}}$ identity matrix. In exact arithmetic $\mathbf{G} = \mathbf{R}^{\mathrm{T}}\mathbf{F} = \mathbf{0}$ and $\tilde{\mathbf{G}} = \mathbf{R}^{\mathrm{T}}$, respectively, which may be used for verification. If $\mathbf{K}$ is stored in sparse form, such as a skyline format, the coefficient matrices of (40) fit the well-known "arrowhead" format as $\mathbf{R}$ generally will couple all equations. However, forward symmetric Gauss elimination will require diagonal pivoting because $\mathbf{K}$ is singular, thus hindering sparsity. Backward elimination can be completed without pivoting, but this will fill out the $\mathbf{K}$ block. Hence exploitation of sparsity with (40) remains troublesome, and the additional programming complexity tilts the balance toward the more compact arrangements (39).

7.2. Woodburying

If N_{F} exceeds roughly 500 freedoms direct inversion becomes progressively expensive in terms of CPU time and storage, and becomes impractical in the thousand-freedom or above range. This would be the

case, for example, when substructures arise as a result of a domain decomposition process for task-parallel processing. Furthermore, since **F** is generally full, storing the complete flexibility would demand significant storage resources. Fortunately, as noted in Section 4, more often than not only a *boundary subset* of **F** needs to be computed, which reduces the need for storage. It is therefore of interest to develop computational techniques that take advantage of

(i) The sparseness of **K**, in the sense that *pivoting on the* **K** *block is precluded.*
(ii) The low rank of **R**.
(iii) The need for only a boundary subset of **F**.

If only RBMs are allowed in **R** this matrix has at most rank 6. It is therefore tempting to consider using the Woodbury inverse-update formula for $(\mathbf{K}+\mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}$. The standard formula cannot be used, however, because **K** is singular. A work-around is developed in Appendix A: the formula is applied to $\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}}$, where **H** is a $N_{\mathrm{F}}\times N_{\mathrm{R}}$ matrix such that $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is a diagonal matrix of rank N_{R} that renders $\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}}$ positive definite while preserving its sparsity and makes pivoting unnecessary.

The necessary theory is worked out in Section A.3 of the Appendix, and only the relevant results are transcribed here. In those equations replace **A** by **K**, **B** by **F**, n by N_{F} and k by N_{R}, while keeping the same notation for other matrices and vectors. The first of (39) is replaced by

$$(\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}}-\mathbf{H}\mathbf{H}^{\mathrm{T}}+\mathbf{R}\mathbf{R}^{\mathrm{T}})\mathbf{F}=\mathbf{P}. \tag{41}$$

The equivalent block-symmetric form is

$$\begin{bmatrix}\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}} & \mathbf{R} & \mathbf{H}\\ \mathbf{R}^{\mathrm{T}} & -\mathbf{I}_{\mathrm{R}} & \mathbf{0}\\ \mathbf{H}^{\mathrm{T}} & \mathbf{0} & \mathbf{I}_{\mathrm{R}}\end{bmatrix}\begin{bmatrix}\mathbf{F}\\ \mathbf{G}_{\mathrm{R}}\\ \mathbf{G}_{\mathrm{H}}\end{bmatrix}=\begin{bmatrix}\mathbf{P}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}. \tag{42}$$

Since $\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is positive definite no pivoting is necessary while processing that block, which contains the bulk of the equations. The explicit matrix borderings of (42) can be avoided by doing two linear blocksolves followed by a matrix combination:

$$\begin{aligned}&(\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}})\mathbf{X}=\mathbf{I},\quad (\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}})\mathbf{Y}_{\mathrm{R}}=\mathbf{R},\\ &\mathbf{F}=\mathbf{P}\mathbf{X}\mathbf{P}=\mathbf{X}-\mathbf{Y}_{\mathrm{R}}\mathbf{R}^{\mathrm{T}}-\mathbf{R}\mathbf{Y}_{\mathrm{R}}^{\mathrm{T}}+\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{Y}_{\mathrm{R}}\mathbf{R}^{\mathrm{T}}.\end{aligned} \tag{43}$$

In the first stage, $\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is factored by a sparse symmetric solver and $N_{\mathrm{F}}+N_{\mathrm{R}}$ right hand sides, namely the columns of **I** and **R**, solved for. The second (combination) stage involves dyadic matrix multiplications, which may be resequenced as $\mathbf{Z}=\mathbf{X}\mathbf{P}=\mathbf{X}-\mathbf{Y}_{\mathrm{R}}\mathbf{R}^{\mathrm{T}}$ followed by $\mathbf{F}=\mathbf{P}\mathbf{Z}=\mathbf{Z}-\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{Z}$. This reorganization is useful when computing a subset $\mathbf{F}_{bb}$ of **F**, as further discussed in Section 7.5.

7.3. Constructing **H**

To close the algorithm specification, it is necessary to state how **H** is constructed. The scheme (43) is valid for an arbitrary **H** matrix such that $\mathbf{H}^{\mathrm{T}}\mathbf{R}$ has full rank. To preserve the sparsity of **K**, however, $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ is required to be diagonal. That requirement is easily handled by choosing the columns of **H** to be *elementary vectors* $\mathbf{h}_j$. This vector has all zero entries except for its jth entry, which is set to $s_j>0$. Obviously $\mathbf{h}_j\mathbf{h}_j^{\mathrm{T}}$ is the null matrix except for the jth diagonal entry, which is s_j^2. Since this is added to K_{jj}, s_j^2 can be viewed as the stiffness of a *penalty spring* added to the jth freedom. The goal is to insert N_{R} springs to suppress the N_{R} rigid body modes. Two insertion strategies may be followed:

1. *Prescribed insertion*. Penalty spring stiffnesses and their locations are picked beforehand from inspection or separate analysis of the substructure.

2. *Adaptive insertion*. The N_{R} penalty springs are inserted "on the fly" as a byproduct of the factorization of **K**, as described below. This strategy has the advantage of providing consistency cross-checks, and is the one selected here.

This kind of adaptive strategy was originally proposed for the stable traversal of critical points in geometrically nonlinear analysis [15,16]. Although there are procedural differences (in critical point traversal usually one spring is sufficient) the underlying idea is the same.

The following tests are patterned after the skyline solver implementation described in [17], but are applicable with minor changes to most direct solvers. We assume that the sparse factorization program carries out the symmetric $\mathrm{LDL}^{\mathrm{T}}$ decomposition

$$\mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}}=\mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}, \tag{44}$$

where **D** is diagonal and **L** unit lower triangular. As pre-processing step the factoring program computes the Euclidean row lengths ℓ_j of **K** and their running maxima m_j, which are saved in a work array:

$$\ell_j=\left(\sum_{i=1}^{N_{\mathrm{F}}}K_{ij}^2\right)^{1/2},\quad m_j=\max_{i=1}^{j}\ell_i,\quad j=1,\ldots,N_{\mathrm{F}}. \tag{45}$$

Initially $\mathbf{H}=\mathbf{0}$. Suppose that the factorization (44) has reached the jth column and row. Entry d_j of **D** (known as a diagonal pivot) is computed as last substep. The following singularity test is performed:

$$|d_j|\leqslant C_d\epsilon m_j=C_{\mathrm{tol}}m_j. \tag{46}$$

Here ϵ is a machine tolerance (the smallest number for which $1+\epsilon>1$ in the floating-point arithmetic used) and C_d is a "loosen up" coefficient typically in the range 10 to 100. If (46) holds, a penalty spring with $s_j^2=C_s m_j$, where C_s is a coefficient of order 10^2 or 10^3, is added to d_j so effectively $d_j\approx s_j^2$. The penalty spring count is incremented by one and the factorization continued.

If **K** is found to be N_{R} times singular as per (46), N_{R} springs are inserted. On exit, the adaptive procedure has effectively changed **K** by the diagonal matrix $\mathbf{H}\mathbf{H}^{\mathrm{T}}$ of rank N_{R} defined as

$$\mathbf{D}_{\mathrm{H}}=\mathbf{H}\mathbf{H}^{\mathrm{T}}=\sum_{i=1}^{N_{\mathrm{R}}}\mathbf{h}_j\mathbf{h}_j^{\mathrm{T}},\quad \text{where } i\text{th entry of } \mathbf{h}_j=s_j>0 \text{ if } i=j,\ \text{else } 0, \tag{47}$$

where **H** is the $N_{\mathrm{F}}\times N_{\mathrm{R}}$ rectangular matrix obtained by stacking the $\mathbf{h}_j$'s as columns. For example, suppose that $N_{\mathrm{F}}=6$, $N_{\mathrm{R}}=2$ and that freedoms $j=3$ and $j=6$ receive penalty springs. Then

$$\mathbf{h}_3=\begin{bmatrix}0\\0\\s_3\\0\\0\\0\end{bmatrix},\quad \mathbf{h}_6=\begin{bmatrix}0\\0\\0\\0\\0\\s_6\end{bmatrix},\quad \mathbf{H}=[\,\mathbf{h}_3\quad\mathbf{h}_6\,]=\begin{bmatrix}0&0\\0&0\\s_3&0\\0&0\\0&0\\0&s_6\end{bmatrix},\quad \mathbf{H}\mathbf{H}^{\mathrm{T}}=\begin{bmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&s_3^2&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&s_6^2\end{bmatrix}. \tag{48}$$

It is important to note that **H** is used in the theoretical developments for convenience but is never formed explicitly.

The foregoing scheme is called an *exact penalty method* in the sense that in exact arithmetic the penalty spring values have no effect on the final result for **F**, as long as they are positive nonzero. In inexact floating-point arithmetic it is wise to constrain the s_j to a certain range to minimize rounding errors and cancellations, but within that range the results remain insensitive to the value. The recommended range $s_j^2=100m_j$ to $1000m_j$ fulfills that objective.

On factorization exit it is necessary to verify whether the penalty spring count is N_{R}. This a posteriori consistency check is discussed in Section 7.9.

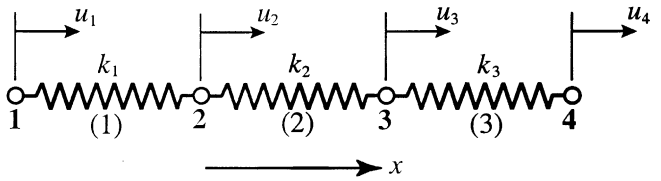

Fig. 8. Three springs in series.

7.4. Tutorial example: Three springs in series

This simple example is worked out in detail to display the equivalence between the direct and penalty-spring flexibility computations. It consists of three springs of stiffnesses k_1, k_2 and k_3 attached in series, as shown in Fig. 8. The springs can only move longitudinally and consequently the substructure has 4 d.o.f.: u_1 through u_4. The stiffness, rigid-body and projector matrices are

$$\mathbf{K} = \begin{bmatrix} k_1 & -k_2 & 0 & 0 \\ -k_2 & k_1+k_2 & -k_2 & 0 \\ 0 & -k_2 & k_2+k_3 & -k_3 \\ 0 & 0 & -k_3 & k_3 \end{bmatrix}, \quad \mathbf{R} = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \quad \mathbf{P} = \frac{1}{4}\begin{bmatrix} 3 & -1 & -1 & -1 \\ -1 & 3 & -1 & -1 \\ -1 & -1 & 3 & -1 \\ -1 & -1 & -1 & 3 \end{bmatrix}. \tag{49}$$

If $k_1 = k_2 = k_3 = 1$, a direct calculation gives

$$\mathbf{F} = \mathbf{P}(\mathbf{K}+\mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1} = (\mathbf{K}+\mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{P} = \frac{1}{8}\begin{bmatrix} 7 & 1 & -3 & -5 \\ 1 & 3 & -1 & -3 \\ -3 & -1 & 3 & 1 \\ -5 & -3 & 1 & 7 \end{bmatrix}. \tag{50}$$

The calculation is now repeated with the method of penalty springs. Because $N_{\mathrm{R}} = 1$ one spring has to be injected. The adaptive collocation method will insert s^2 at freedom u_4 because the symmetric forward factorization of $\mathbf{K}$ will get a zero or tiny pivot there. Thus

$$\mathbf{H} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ s \end{bmatrix}, \quad \mathbf{K}_{\mathrm{H}} = \mathbf{K}+\mathbf{H}\mathbf{H}^{\mathrm{T}} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1+s^2 \end{bmatrix}. \tag{51}$$

We will keep s arbitrary to study its propagation in the calculation sequence. Solving the systems $\mathbf{K}_{\mathrm{H}}\mathbf{X} = \mathbf{I}$ and $\mathbf{K}_{\mathrm{H}}\mathbf{Y}_{\mathrm{R}} = \mathbf{R}$, we obtain

$$\mathbf{X} = \frac{1}{4}\begin{bmatrix} 3+1/s^2 & 2+1/s^2 & 1+1/s^2 & 1/s^2 \\ 2+1/s^2 & 2+1/s^2 & 1+1/s^2 & 1/s^2 \\ 1+1/s^2 & 1+1/s^2 & 1+1/s^2 & 1/s^2 \\ 1/s^2 & 1/s^2 & 1/s^2 & 1/s^2 \end{bmatrix}, \quad \mathbf{Y}_{\mathrm{R}} = \frac{1}{2}\begin{bmatrix} 6+4/s^2 \\ 5+4/s^2 \\ 3+4/s^2 \\ 4/s^2 \end{bmatrix}, \tag{52}$$

and combining

$$\mathbf{F} = \mathbf{X} - \mathbf{Y}_{\mathrm{R}}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}} - \mathbf{R}\mathbf{Y}_{\mathrm{R}}^{\mathrm{T}} - \mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{Y}_{\mathrm{R}}\mathbf{R}^{\mathrm{T}} = \frac{1}{8}\begin{bmatrix} 7 & 1 & -3 & -5 \\ 1 & 3 & -1 & -3 \\ -3 & -1 & 3 & 1 \\ -5 & -3 & 1 & 7 \end{bmatrix}. \tag{53}$$

As expected the effect of a nonzero s cancels out in $\mathbf{F}$ because of the exact arithmetic. Numerical computations in double-precision floating-point show that $\mathbf{F}$ is obtained with full (16-place) accuracy if $s > 1$

(but not so large as to cause overflow). If $s < 1$, the accuracy in $\mathbf{X}$ drops because $\mathbf{K} + \mathbf{H}\mathbf{H}^{\mathrm{T}}$ approaches a singular matrix, since its condition number is $C(\mathbf{K}_{\mathrm{H}}) \approx 13.6/s^2$ for $s \ll 1$.

7.5. Computing a subset of $\mathbf{F}$

As noted in Section 4, for many applications of the free–free flexibility it is only necessary to evaluate a submatrix $\mathbf{F}_b \equiv \mathbf{F}_{bb}$ of $\mathbf{F}$ that pertains to $N_b < N_{\mathrm{F}}$ freedoms at boundary nodes. The penalty spring procedure can be adjusted to cut down on both computations and storage. The changes are best illustrated through a modification of the previous example. Suppose that only the end freedoms (u_1, f_1) and (u_4, f_4) are of interest for $\mathbf{F}_{bb}$. The construction of $\mathbf{K}$, $\mathbf{H}$ and $\mathbf{K}_{\mathrm{H}} = \mathbf{K} + \mathbf{H}\mathbf{H}^{\mathrm{T}}$ proceeds as before and the solution for $\mathbf{Y}_{\mathrm{R}}$ does not change. The linear system $\mathbf{K}_{\mathrm{H}}\mathbf{X} = \mathbf{I}$ may be reduced to $\mathbf{K}_{\mathrm{H}}\mathbf{X}_b = \mathbf{I}_b$, in which the 4×2 matrices $\mathbf{X}_b$ and $\mathbf{I}_b$ hold columns 1 and 4 of $\mathbf{X}$ and $\mathbf{I}$, respectively. A trimmed version $\mathbf{R}_b$ of the $\mathbf{R}$ matrix that keeps only rows 1 and 4 is formed. For this example

$$\mathbf{X}_b = \frac{1}{4}\begin{bmatrix} 3+1/s^2 & 1/s^2 \\ 2+1/s^2 & 1/s^2 \\ 1+1/s^2 & 1/s^2 \\ 1/s^2 & 1/s^2 \end{bmatrix}, \quad \mathbf{Y}_{\mathrm{R}} = \frac{1}{2}\begin{bmatrix} 6+4/s^2 \\ 5+4/s^2 \\ 3+4/s^2 \\ 4/s^2 \end{bmatrix}, \quad \mathbf{R} \to \mathbf{R}_b = \frac{1}{2}\begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{54}$$

The $N_{\mathrm{F}} \times N_b$ matrix $\mathbf{Z}_b = \mathbf{X}_b - \mathbf{Y}_{\mathrm{R}}\mathbf{R}_b^{\mathrm{T}}$ is computed, and a trimmed version $\mathbf{Z}_{bb}$ formed by keeping only rows 1 and 4:

$$\mathbf{Z}_b = \mathbf{X}_b - \mathbf{Y}_{\mathrm{R}}\mathbf{R}_b^{\mathrm{T}} = \frac{1}{4}\begin{bmatrix} 6 & -6 \\ 3 & -5 \\ 1 & -3 \\ 0 & 0 \end{bmatrix} \to \mathbf{Z}_{bb} = \frac{1}{4}\begin{bmatrix} 6 & -6 \\ 0 & 0 \end{bmatrix}. \tag{55}$$

Combining:

$$\mathbf{F}_{bb} = \mathbf{Z}_{bb} - \mathbf{R}_b\mathbf{R}^{\mathrm{T}}\mathbf{Z}_b = \frac{1}{8}\begin{bmatrix} 7 & -5 \\ -5 & 7 \end{bmatrix}. \tag{56}$$

The projected boundary flexibility is

$$\bar{\mathbf{F}}_b = \mathbf{P}_b\mathbf{F}_{bb}\mathbf{P}_b = \frac{3}{4}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix},$$

in which $\mathbf{P}_b = \mathbf{I}_2 - \mathbf{R}_b(\mathbf{R}_b^{\mathrm{T}}\mathbf{R}_b)^{-1}\mathbf{R}_b^{\mathrm{T}}$. This is the pseudo-inverse of the condensed stiffness

$$\mathbf{K}_b = \frac{1}{3}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

For this simple problem these gyrations are hardly worth the trouble. But they make sense for larger systems. For example, a substructure made of a 50×50 plane stress quadrilateral mesh contains 400 boundary and 4802 internal freedoms, respectively. Hence $\mathbf{F}$ would be 5202×5202, requiring 210 MB for storage in double precision as a full matrix. But the 400×400 matrix $\mathbf{F}_{bb}$ would need only 1.3 MB.

7.6. A model reduction example

This example illustrates the application of $\mathbf{F}$ to a model reduction process that retains only selected boundary freedoms. It can also be used as a benchmark problem to test implementations of the exact penalty spring method. Fig. 9(a) shows a 16-element, free–free substructure built with 4-node square plane stress elements with two freedoms per node. The mesh has 25 nodes and $N_{\mathrm{F}} = 50$ d.o.f., and three independent RBMs ($N_{\mathrm{R}} = 3$). The plate dimensions and thickness are shown in the figure. The material is

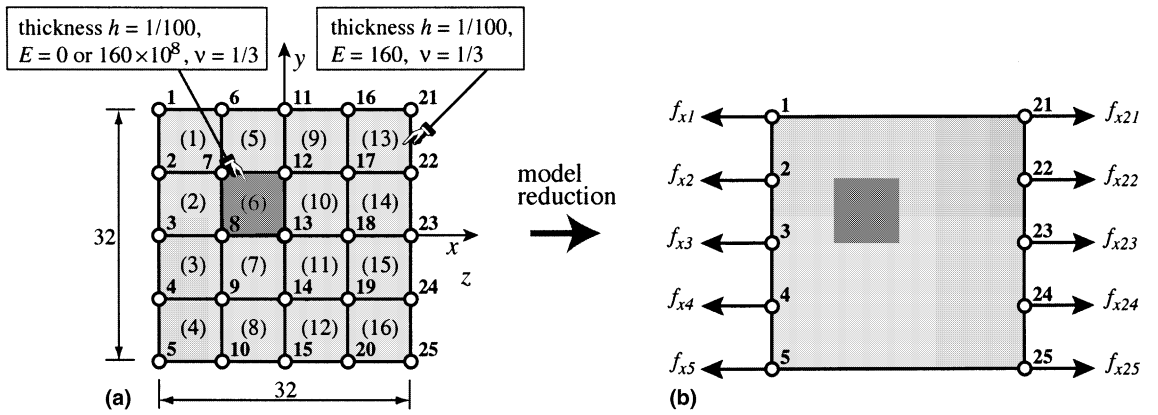

Fig. 9. Application of **F** to model reduction: (a) a 25-node, 50-DOF plane-stress substructure; (b) a reduced model that retains only 10 boundary freedoms. The shaded element (6) models either a hole ($E^{(6)} = 0$) or a near-rigid inclusion ($E^{(6)} = 160 \times 10^8$).

isotropic with $E = 160$ and $v = 1/3$ except for element 6, which is taken to have either $E^{(6)} = 0$ (hole) or $E^{(6)} = 160 \times 10^8$ (near-rigid inclusion). The reduced model retains only the 10 horizontal displacements and forces at the boundary nodes shown in Fig. 9(b). (There are 32 boundary freedoms, but only 10 are kept here to keep $\mathbf{F}_{bb}$ printable.)

Although the problem size is too small for sparse computations to be advantageous, it facilitates the visualization of matrix configurations as depicted in Fig. 10.

With the $\{x, y\}$ axes placed as shown, the 50×3 orthonormalized RBM matrix **R** is

$$\mathbf{R}^{\mathrm{T}} = \frac{1}{10}\begin{bmatrix} 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & \dots & 2 & 0 \\ 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & 0 & 2 & \dots & 0 & 2 \\ -2 & -2 & -1 & -2 & 0 & -2 & 1 & -2 & 2 & -2 & -2 & -1 & \dots & 2 & 2 \end{bmatrix}. \tag{57}$$

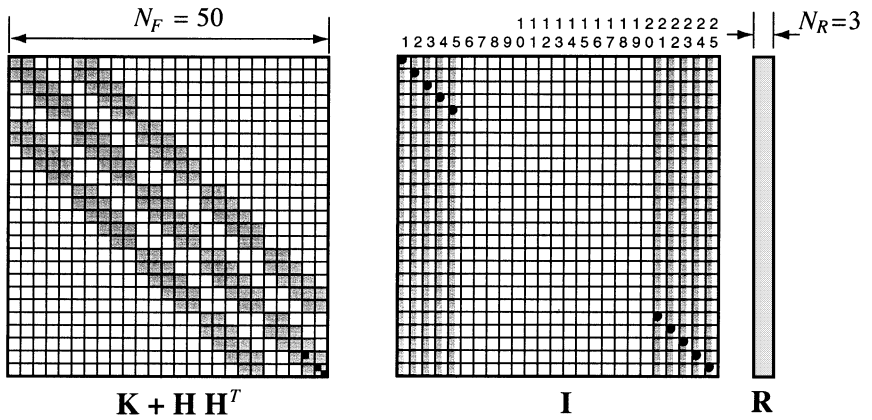

Fig. 10. Leftmost diagram shows the configuration of $\mathbf{K} + \mathbf{H}\mathbf{H}^{\mathrm{T}}$ for the substructure of Fig. 9. Each square represents a 2×2 block. White squares mark zero entries. The rightmost diagrams depict the configuration of the right-hand side matrices **I** and **R**. Border numbers above **I** are node numbers. Shaded columns in **I** mark the 10 columns of **I** to be processed. Black circles mark the unit diagonal elements at which processing starts. Three penalty springs suppressing the RBMs are inserted at freedoms 47, 49 and 50. These are marked with black squares in the diagonal of **K**.

The isoparametric element stiffness matrix for $\nu = 1/3$, $h = 1/100$ and variable E is

$$\mathbf{K}^{(e)} = \frac{E}{1600}\begin{bmatrix} 8 & 3 & -5 & 0 & -4 & -3 & 1 & 0 \\ 3 & 8 & 0 & 1 & -3 & -4 & 0 & -5 \\ -5 & 0 & 8 & -3 & 1 & 0 & -4 & 3 \\ 0 & 1 & -3 & 8 & 0 & -5 & 3 & -4 \\ -4 & -3 & 1 & 0 & 8 & 3 & -5 & 0 \\ -3 & -4 & 0 & -5 & 3 & 8 & 0 & 1 \\ 1 & 0 & -4 & 3 & -5 & 0 & 8 & -3 \\ 0 & -5 & 3 & -4 & 0 & 1 & -3 & 8 \end{bmatrix}, \tag{58}$$

from which the 50×50 free–free stiffness matrix $\mathbf{K}$ is easily assembled by the DSM.

For the hole case, with $E^{(6)} = 0$ and $E = 160$ elsewhere, the diagonal entries of $\mathbf{K}$ are

$$\operatorname{diag}(\mathbf{K}) = \tfrac{1}{5}[4\ 4\ 8\ 8\ 8\ 8\ 8\ 8\ 4\ 4\ 8\ 8\ 12\ 12\ 12\ 12\ 16\ 16\ 8\ 8\ \cdots\ 8\ 8\ 4\ 4]. \tag{59}$$

The rowlength maxima, listed to three places, are

$$m_j = [1.11\ 1.11\ 2.02\ 2.02\ 2.02\ 2.02\ 2.02\ 2.02\ 2.02\ 2.02\ 2.02\ 2.02\ 2.81\ 2.81\ 2.81\ 2.81\ 3.65\ 3.65 \cdots 3.65]. \tag{60}$$

On factoring $\mathbf{K} = \mathbf{L}\mathbf{D}\mathbf{L}^{\mathrm{T}}$ in double-precision floating-point the first 47 diagonal entries of $\mathbf{D}$, listed to four decimal places, are

$$\operatorname{diag}(\mathbf{D})_{|1:47|} = \left|\, 0.8000 \quad 0.6875 \quad 1.5854 \quad 1.2368 \quad \cdots \quad 0.1851 \quad 0.7415 \quad 1.066 \times 10^{-14} \,\right|. \tag{61}$$

With $C_{\mathrm{tol}} = 10^{-13}$ used in the test (46), the first singularity is detected at $j = 47$ because $d_{47} = 1.066 \times 10^{-14} < 10^{-13} \times 3.65$. A penalty spring $s_{47}^2 = 100 \times 3.65 = 365$ is inserted, whence d_{47} becomes s_{47}^2 to 14 places. The factorization continues with $d_{48} = 0.6084$ and $d_{49} = 9.326 \times 10^{-15}$. The second spring is inserted at $j = 49$ changing d_{49} also to 365. The third singularity is detected at the last entry: $d_{50} = 9.992 \times 10^{-16}$. The third spring goes at $j = 50$ and d_{50} becomes 365. On completing the remaining steps in (43) the following reduced flexibility, listed to 4 decimal places, is obtained:

$$\mathbf{F}_{bb} = \begin{bmatrix} 2.0510 & 0.0587 & -0.2545 & -0.2086 & -0.0018 & -0.4334 & -0.2064 & -0.1644 & -0.1145 & 0.0295 \\ & 1.0558 & 0.1168 & -0.1083 & -0.1746 & -0.1971 & -0.1462 & -0.1680 & -0.1518 & -0.0843 \\ & & 1.0647 & 0.1420 & -0.2225 & -0.2334 & -0.1676 & -0.1172 & -0.0846 & -0.0894 \\ & & & 0.9642 & -0.0260 & -0.1486 & -0.1274 & -0.0843 & -0.0808 & -0.1920 \\ & & & & 1.9373 & 0.0950 & -0.0489 & -0.1240 & -0.2237 & -0.5066 \\ & & & & & 2.0534 & 0.0313 & -0.2078 & -0.1960 & -0.0556 \\ & & & & & & 0.9341 & 0.0831 & -0.0797 & -0.1731 \\ & & & & & & & 0.9442 & 0.0934 & -0.1841 \\ & & & & & & & & 0.9313 & 0.0080 \\ \text{symmetric} & & & & & & & & & 1.9614 \end{bmatrix}. \tag{62}$$

Its eigenvalues are

$$[\,2.6335 \quad 2.4016 \quad 2.0182 \quad 1.4903 \quad 1.4570 \quad 1.0796 \quad 0.9497 \quad 0.7887 \quad 0.7717 \quad 0.3071\,].$$

Comparison with the exact full $\mathbf{F}$ obtained in rational arithmetic with *Mathematica* shows that the double-precision floating-point computation sequence delivers 15 places of accuracy.

For the near-rigid intrusion case, in which $E^{(6)} = 160 \times 10^8$ and $E = 160$ elsewhere, the diagonal entries of $\mathbf{K}$ are as in (59) except at nodes 7, 8, 12 and 13 ($j = 13, 14, 15, 16, 23, 24, 25, 26$) where they jump to

8000002.4. The rowlength maxima m_j start as in (60), but jump to 1.1136×10^7 at $j = 13$ and stay at that value through $j = 50$. The factorization proceeds without incident until reaching $d_{47} = 2.19 \times 10^{-8}$, which flags a singularity since $2.19 \times 10^{-8} < 10^{-13} \times 1.1136 \times 10^7$. A penalty spring of $s^2_{47} = 100 \times 1.1136 \times 10^7 = 1.1136 \times 10^9$ is added there. The next tiny pivot is $d_{49} = 1.028 \times 10^{-8}$, immediately followed by $d_{50} = 1.3175 \times 10^{-12}$. The same penalty spring values are inserted at $j = 49$ and $j = 50$. Proceeding with the remaining steps the following reduced flexibility, listed to four decimal places, is obtained:

$$\mathbf{F}_{bb} = \begin{bmatrix} 1.8371 & -0.0633 & -0.1338 & -0.1101 & -0.0138 & -0.5129 & -0.2392 & -0.1377 & -0.0480 & 0.1181 \\ & 0.7467 & -0.0018 & -0.0542 & -0.1238 & -0.2237 & -0.0675 & -0.0323 & -0.0559 & -0.0674 \\ & & 0.7546 & 0.0068 & -0.1519 & -0.1007 & -0.0346 & -0.0250 & -0.0799 & -0.1762 \\ & & & 0.8500 & -0.0041 & -0.0321 & -0.0725 & -0.0898 & -0.1230 & -0.2503 \\ & & & & 1.9126 & 0.0859 & -0.0832 & -0.1618 & -0.2379 & -0.4886 \\ & & & & & 1.8441 & -0.0470 & -0.1767 & -0.1204 & 0.0183 \\ & & & & & & 0.8622 & 0.0396 & -0.0776 & -0.1281 \\ & & & & & & & 0.8632 & 0.0398 & -0.1830 \\ & & & & & & & & 0.8729 & -0.0270 \\ \text{symmetric} & & & & & & & & & 1.9030 \end{bmatrix}. \tag{63}$$

Its eigenvalues are

$$[2.5252 \quad 2.2875 \quad 1.8236 \quad 1.2890 \quad 1.0172 \quad 0.9076 \quad 0.8070 \quad 0.7698 \quad 0.7294 \quad 0.2902].$$

Comparison with the exact **F** obtained in rational arithmetic shows that the double-precision floating-point computation delivers on average 11 places of accuracy. Hence the effect of the poorly scaled stiffness has been to lose four decimal places with respect to the well-scaled (hole) case.

Although the full stiffness **K** and flexibility **F** of the two cases have entries that differ by eight orders of magnitude, the reduced-model flexibilities are of the same order. Furthermore, observe that both (62) and (63) are *diagonally dominant and very well conditioned*. This ability of boundary flexibilities to filter out interior details is a key factor in the success of FETI-type iterative solvers [1–4].

7.7. A substructure with internal mechanisms

The last example illustrates how to handle a substructure that exhibits non-RBM zero energy modes. Symbolic analysis is used to make the result valid for a wide range of property data and hence more useful as validation benchmark. Three identical prismatic plane beam-column members are connected as shown in Fig. 11(a). The hinge at node 4 permits members to rotate independently: $\theta_4^{(1)} \neq \theta_4^{(2)} \neq \theta_4^{(3)}$, while enforcing equal translations. The system has the 14 d.o.f. defined in Fig. 11(a), which are ordered as

$$\mathbf{u}^{\mathrm{T}} = \left[u_{x1} \quad u_{y1} \quad L\theta_1 \quad u_{x2} \quad u_{y2} \quad L\theta_2 \quad u_{x3} \quad u_{y3} \quad L\theta_3 \quad u_{x4} \quad u_{y4} \quad L\theta_4^{(1)} \quad L\theta_4^{(2)} \quad L\theta_4^{(3)} \right]. \tag{64}$$

The rotational freedoms are scaled by L to get dimensionally homogeneous results. The three members have the same length L, elastic modulus E, cross-section area A and moment of inertia $I = I_{zz}$. For convenience in subsequent symbolic expressions, we set $I = r^2A = \rho^2AL^2$, where r is the cross-section radius of gyration and $\rho = r/L$ the reciprocal of the bending slenderness ratio. The assembled free–free stiffness is

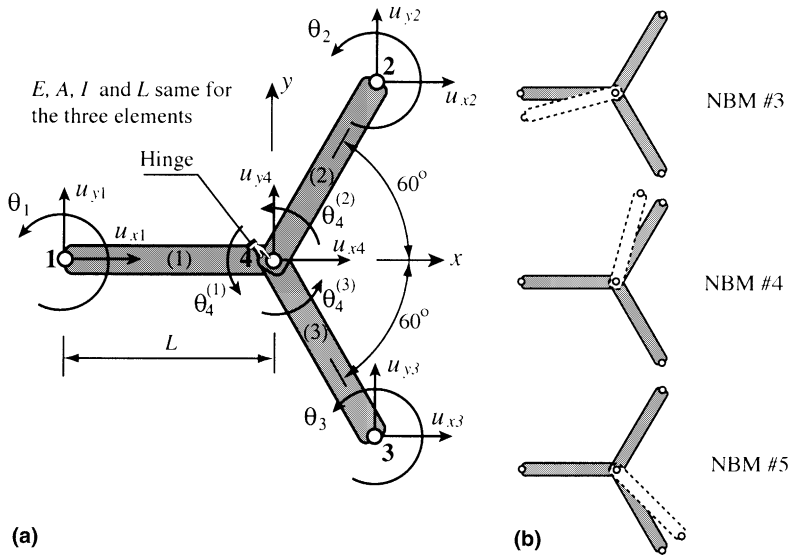

Fig. 11. Three identical plane beam-columns hinged at node 4. (b) depicts three null basis modes (NBM) used to start the construction of the 5×14 null space basis matrix (66); NBMs #1 and #2 are the translational RBMs.

$$\mathbf{K} = \frac{EA}{L}\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 12\rho^2 & 6\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -12\rho^2 & 6\rho^2 & 0 & 0 \\
0 & 6\rho^2 & 4\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6\rho^2 & 2\rho^2 & 0 & 0 \\
0 & 0 & 0 & \mu_2 & \mu_1 & \mu_5 & 0 & 0 & 0 & -\mu_2 & -\mu_1 & 0 & \mu_5 & 0 \\
0 & 0 & 0 & \mu_1 & \mu_3 & -3\rho^2 & 0 & 0 & 0 & -\mu_1 & -\mu_3 & 0 & -3\rho^2 & 0 \\
0 & 0 & 0 & \mu_5 & -3\rho^2 & 4\rho^2 & 0 & 0 & 0 & -\mu_5 & 3\rho^2 & 0 & 2\rho^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & \mu_2 & -\mu_1 & -\mu_5 & -\mu_2 & \mu_1 & 0 & 0 & -\mu_5 \\
0 & 0 & 0 & 0 & 0 & 0 & -\mu_1 & \mu_3 & -3\rho^2 & \mu_1 & -\mu_3 & 0 & 0 & -3\rho^2 \\
0 & 0 & 0 & 0 & 0 & 0 & -\mu_5 & -3\rho^2 & 4\rho^2 & \mu_5 & 3\rho^2 & 0 & 0 & 2\rho^2 \\
-1 & 0 & 0 & -\mu_2 & -\mu_1 & -\mu_5 & -\mu_2 & \mu_1 & \mu_5 & \mu_4 & 0 & 0 & -\mu_5 & \mu_5 \\
0 & -12\rho^2 & -6\rho^2 & -\mu_1 & -\mu_3 & 3\rho^2 & \mu_1 & -\mu_3 & 3\rho^2 & 0 & \mu_4 & -6\rho^2 & 3\rho^2 & 3\rho^2 \\
0 & 6\rho^2 & 2\rho^2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6\rho^2 & 4\rho^2 & 0 & 0 \\
0 & 0 & 0 & \mu_5 & -3\rho^2 & 2\rho^2 & 0 & 0 & 0 & -\mu_5 & 3\rho^2 & 0 & 4\rho^2 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -\mu_5 & -3\rho^2 & 2\rho^2 & \mu_5 & 3\rho^2 & 0 & 0 & 4\rho^2
\end{bmatrix}, \tag{65}$$

in which

$$\mu_1 = \tfrac{1}{4}\sqrt{3}(1 - 12\rho^2), \quad \mu_2 = \tfrac{1}{4} + 9\rho^2, \quad \mu_3 = \tfrac{3}{4} + 3\rho^2, \quad \mu_4 = \tfrac{3}{2} + 18\rho^2 \quad \text{and} \quad \mu_5 = 3\sqrt{3}\rho^2.$$

The 5×14 null space basis will also be denoted by $\mathbf{R}$. This matrix contains the three rigid-body modes plus two mechanisms, and can be constructed directly by geometric inspection without having to analyze $\mathbf{K}$. As initial basis we select the two translational RBMs along x and y, plus the three individual beam rotations depicted in Fig. 11(b). Applying Gram–Schmid yields the orthonormal basis

$$\mathbf{R}^{\mathrm{T}} = \begin{bmatrix} \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & -3\gamma_1 & 4\gamma_1 & 0 & \gamma_1 & 0 & 0 & \gamma_1 & 0 & 0 & \gamma_1 & 4\gamma_1 & 0 & 0 \\ \gamma_2 & -2\gamma_3 & -\gamma_3 & -3\gamma_2 & 8\gamma_3 & \gamma_4 & \gamma_2 & -2\gamma_5 & 0 & \gamma_2 & -2\gamma_5 & -\gamma_3 & \gamma_4 & 0 \\ -6\gamma_7 & -4\gamma_6 & -2\gamma_6 & -5\gamma_7 & -7\gamma_6 & -2\gamma_6 & 17\gamma_7 & 17\gamma_6 & 46\gamma_6 & -6\gamma_7 & -6\gamma_6 & -2\gamma_6 & -2\gamma_6 & \gamma_8 \end{bmatrix}, \tag{66}$$

in which

$$\gamma_1 = 1/(2\sqrt{11}), \quad \gamma_2 = \tfrac{1}{2}\sqrt{11/161}, \quad \gamma_3 = 2/\sqrt{5313}, \quad \gamma_4 = 4\sqrt{11/483}, \quad \gamma_5 = \sqrt{3/1771},$$
$$\gamma_6 = 1/(6\sqrt{161}), \quad \gamma_7 = 1/(2\sqrt{483}) \quad \text{and} \quad \gamma_8 = \sqrt{23/63}.$$

A symbolic calculation with *Mathematica* using the method of penalty springs delivers the following boundary flexibility associated with the 9 d.o.f. at nodes 1, 2 and 3:

$$\mathbf{F}_{bb} = \frac{L}{5292EA\rho^2} \begin{bmatrix} \chi_1 & 0 & 0 & -\chi_6 & \chi_9 & -\chi_{10} & -\chi_6 & -\chi_9 & \chi_{10} \\ 0 & \chi_2 & \chi_7 & 2\chi_8 & -\chi_{11} & \chi_{12} & -2\chi_8 & -\chi_{11} & \chi_{12} \\ 0 & \chi_7 & \chi_3 & \chi_8 & -\chi_{13} & \chi_{14} & -\chi_8 & -\chi_{13} & \chi_{14} \\ -\chi_6 & 2\chi_8 & \chi_8 & \chi_4 & -\chi_{15} & \chi_{16} & -\chi_{17} & \chi_{18} & \chi_{19} \\ \chi_9 & -\chi_{11} & -\chi_{13} & -\chi_{15} & \chi_5 & -\chi_{20} & -\chi_{18} & -\chi_{21} & \chi_{22} \\ -\chi_{10} & \chi_{12} & \chi_{14} & \chi_{16} & -\chi_{20} & \chi_3 & -\chi_{19} & \chi_{22} & \chi_{14} \\ \chi_6 & -2\chi_8 & -\chi_8 & -\chi_{17} & -\chi_{18} & -\chi_{19} & \chi_4 & \chi_{15} & -\chi_{16} \\ -\chi_9 & -\chi_{11} & -\chi_{13} & \chi_{18} & -\chi_{21} & \chi_{22} & \chi_{15} & \chi_5 & -\chi_{20} \\ \chi_{10} & \chi_{12} & \chi_{14} & \chi_{19} & \chi_{22} & \chi_{14} & -\chi_{16} & -\chi_{20} & \chi_3 \end{bmatrix}, \tag{67}$$

in which

$$\chi_1 = 24 + 2916\rho^2, \quad \chi_2 = 132 + 288\rho^2, \quad \chi_3 = 1356 + 72\rho^2, \quad \chi_4 = 105(1 + 9\rho^2), \quad \chi_5 = 51 + 2259\rho^2,$$
$$\chi_6 = 6(4 + 171\rho^2), \quad \chi_7 = 66 + 144\rho^2, \quad \chi_8 = 3\sqrt{3}(1 - 36\rho^2), \quad \chi_9 = 2\sqrt{3}(8 - 225\rho^2),$$
$$\chi_{10} = 2\sqrt{3}(5 + 72\rho^2), \quad \chi_{11} = 46 + 360\rho^2, \quad \chi_{12} = 16 - 72\rho^2, \quad \chi_{13} = 23 + 180\rho^2, \quad \chi_{14} = 8 - 36\rho^2,$$
$$\chi_{15} = 9\sqrt{3}(3 - 73\rho^2), \quad \chi_{16} = 3\sqrt{3}(11 + 24\rho^2), \quad \chi_{17} = 3(19 + 9\rho^2), \quad \chi_{18} = \sqrt{3}(5 - 117\rho^2),$$
$$\chi_{19} = \sqrt{3}(13 + 36\rho^2), \quad \chi_{20} = 33 + 72\rho^2, \quad \chi_{21} = 13 + 1359\rho^2 \quad \text{and} \quad \chi_{22} = 7 + 252\rho^2.$$

Symbolic penalty springs are inserted at **K**-factorization pivots 10, 11, 12, 13 and 14, which become exact zeroes in the exact arithmetic used by *Mathematica*.

Matrix (67) has full rank of 9 if $\rho > 0$. If $\rho \to \infty$, which for fixed A and L means that the three members become infinitely stiff in bending, $\mathbf{F}_{bb}$ approaches a finite matrix of rank 3 whereas $\mathbf{K} \to \infty$.

7.8. Computational costs

If a sparse skyline solver is used for the factor and blocksolve stages, the amount of arithmetic work required in the penalty spring method may be estimated as follows. Denote by b the root-mean-square bandwidth of **K**, $n = N_{\mathrm{F}}$ the order of **K** and $k = N_{\mathrm{R}}$ the RBM count. The cost in floating-point operation units to get the full **F** is approximately

$$C_{\mathrm{tot}} = C_F + C_S + C_C, \quad C_F = \tfrac{1}{2}nb^2, \quad C_S = 2nb\left(\tfrac{1}{2}n + k\right), \quad C_S = 3n^2k, \tag{68}$$

Table 1
Computational costs and storage demands for sample 2D/3D regular meshes

Case	N	$n = N_{\mathrm{F}}$	n_b	C_F	C_S	C_C	C_{tot}	S in GB
2D $\rightarrow$ **F**	50	5000	5000	0.03	2.50	0.22	2.75	0.10
2D $\rightarrow$ **F**	100	20 000	20 000	0.40	80.02	3.60	84.02	1.63
2D $\rightarrow$ **F**	200	80 000	80 000	6.40	2560.20	57.60	2624.20	25.86
2D $\rightarrow$ $\mathbf{F}_{bb}$	50	5000	400	0.03	0.39	0.02	0.44	0.02
2D $\rightarrow$ $\mathbf{F}_{bb}$	100	20 000	800	0.40	6.30	0.14	6.84	0.16
2D $\rightarrow$ $\mathbf{F}_{bb}$	200	80 000	1600	6.40	101.57	1.15	109.12	1.29
3D $\rightarrow$ **F**	10	3000	3000	0.2	2.7	0.2	3.1	0.04
3D $\rightarrow$ **F**	20	24 000	24 000	17.3	691.6	10.4	719.2	2.54
3D $\rightarrow$ **F**	40	192 000	192 000	2211.8	176958.4	663.6	179833.8	154.85
3D $\rightarrow$ $\mathbf{F}_{bb}$	10	3000	1800	0.2	2.3	0.1	2.6	0.06
3D $\rightarrow$ $\mathbf{F}_{bb}$	20	24 000	7200	17.3	352.9	3.1	373.2	1.82
3D $\rightarrow$ $\mathbf{F}_{bb}$	40	192 000	28 800	2211.8	49113.9	99.6	51425.3	54.95

C's in billions of floating-point operation units, storage in gigabytes.

where C_F, C_S, and C_C denote the costs of $\mathbf{K}_{\mathrm{H}}$-factorization, blocksolve and matrix combination, respectively. For C_S the special form of the right-hand side in block solving $\mathbf{K}_{\mathrm{H}}\mathbf{X} = \mathbf{I}$ has been accounted for. The storage requirement is approximately $S = nb + \frac{1}{2}n^2 + 2nk$ double-precision floating-point (8-byte) locations.

If only a subset $\mathbf{F}_{bb}$ or order $n_b < n$ is required, then again $C_{\mathrm{tot}} = C_F + C_S + C_C$, where now

$$C_F = \tfrac{1}{2}nb^2, \quad C_S = 2nb\left[\left(1 - \frac{n_b}{2n}\right)n_b + k\right], \quad C_S = 3nn_bk. \tag{69}$$

Thus C_F stays the same whereas C_S and C_C drop. The storage required is approximately $S = nb + nn_b + \frac{1}{2}n_b^2 + 2nk$ double-precision floating-point (8-byte) locations.

To provide a more concrete cost picture, two regular mesh configurations are chosen and results posted in Table 1. The case labeled "2D" pertains to a regular $N \times N$ 2D plane stress mesh of 4-node quadrilaterals with two freedoms per node, $n = N_{\mathrm{F}} \approx 2N^2$, $b \approx 2N$, $k = N_{\mathrm{R}} = 3$ and $n_b = 8N$. Table 1 lists costs (in billions of operation units) for $N = 50$, 100 and 200 to compute the full **F** and the boundary-reduced $\mathbf{F}_{bb}$. The leading term in C_{tot} is $8N^5$ for **F** and $68N^4$ for $\mathbf{F}_{bb}$. In both cases the blocksolve cost C_S is dominant. On a one-gigaflop (1 GF) CPU, typical of present high-end offerings by Intel and AMD, the estimate runtime in seconds will be that shown in Table 1 for C_{tot}; for instance 109 s to get the $\mathbf{F}_{bb}$ of a 200×200 mesh.

The case labeled "3D" pertains to a regular $N \times N \times N$ 3D mesh of 8-node bricks with three freedoms per node, $n = N_{\mathrm{F}} \approx 3N^3$, $b \approx 3N^2$, $k = N_{\mathrm{R}} = 6$ and $n_b \approx 18N^2$. The leading term in C_{tot} is $27N^8$ for **F** and $338N^7$ for $\mathbf{F}_{bb}$. Table 1 lists data for $N = 10$, 20 and 40. The total cost is again strongly dominated by the blocksolve. On a 1 GF processor, the computation of $\mathbf{F}_{bb}$ for a $20 \times 20 \times 20$ mesh would take 373 s. Note that the ratio of full to boundary-reduced costs is not so significant as in the 2D case because a larger percentage of nodes are on the boundary; for example, that ratio is only about 2 for $N = 20$.

Several conclusions emerge from the cost analysis:

1. The *factorization cost is comparatively insignificant for fine meshes in 2D and 3D*. Multiple refactorization passes, for example, to test the effect of different singularity tolerances on the penalty spring count, would have minor impact on the total cost.
2. Improved computational efficiency in the blocksolve, for example, using assembly language to squeeze high performance out of superscalar or pipeline processors, would have a more immediate payoff than improvements in the sparse factorization. For the same reason, complicated sparse solvers with high indexing overhead should be avoided.

3. The estimated C_S may be used as a guide in model decomposition to attain load balance in parallel computations where one or more substructures are assigned to each processor. The solution cost will also dominate the parallel recovery of internal states if the factorization of $\mathbf{K}_{\mathrm{H}}$ is saved in each processor.

It should be noted that the cost of forming the projected boundary flexibility $\bar{\mathbf{F}}_b = \mathbf{P}_b\mathbf{F}_{bb}\mathbf{P}_b$ described in Section 4 is not included in these estimates. This may be viewed as a postprocessing step required for some applications.

7.9. Implementation considerations

The determination of rank in floating-point arithmetic is a delicate "ϵ threshold" problem [10,18]. Accordingly, when processing general substructures the penalty spring method should be complemented with appropriate safeguards to achieve robustness. A software configuration that implements two consistency tests is diagramed in Fig. 12.

Upon forming (or procuring) **K** and **R**, an obvious check is whether $\mathbf{KR} = \mathbf{0}$ within floating-point tolerance. Failure flags serious inconsistencies; for example, **K** may come from a "black box" source such as a commercial FEM code and be RBM polluted, or the substructure geometric data are flawed. An immediate error exit is indicated.

The second test checks whether the penalty spring count N_s returned by the sparse factor agrees with the column dimension N_{R} of **R**. If so, processing continues with the blocksolve and combination steps to return either **F** or $\mathbf{F}_{bb}$. Handling discrepancies is tricky because many error sources, whether model-based or

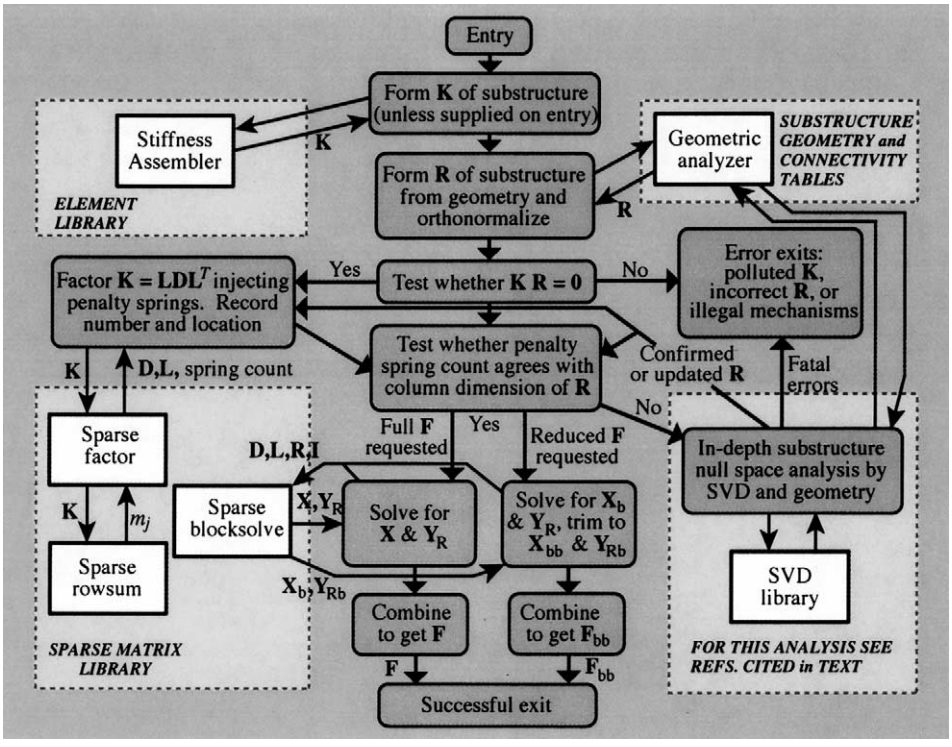

Fig. 12. Schematics of the free–free flexibility analysis of a substructure. For the in-depth null space analysis, which is not treated here, see [19,20].

computational, are possible. A spring count exceeding N_R may be due to (i) spurious kinematic mechanisms, (ii) highly ill-conditioned **K**, or (iii) overly loose tolerance in the singularity test (46). A count less than N_R may be due to (iv) overstrict singularity test tolerance, (v) RBM polluted stiffness, or (vi) physical mechanisms (such as initial stress effects in geometrically nonlinear analysis) that eliminate some RBMs. (The last two sources, however, should be caught by the prior $\mathbf{KR} = \mathbf{0}$? test.)

If a count discrepancy occurs, it is recommended to proceed to an in-depth diagnosis of the null space of **K**, as flowcharted in Fig. 12. This may involve an SVD analysis [19] or a geometric analysis complemented by the SVD [20]. The interested reader should consult those references for procedural details. On return from this step a decision must be made to proceed, or to take an error exit requiring further instructions from the user. Two scenarios are:

1. The null-space analyzer confirms the initial **R**. The factorization singularity tolerance should be then adjusted until the correct number of springs is returned (discrepancy one way or the other is not acceptable and will result in an incorrect flexibility). As noted in the cost study, the expense of multiple refactorizations is negligible for fine 2D and 3D meshes. If agreement cannot be achieved **K** is likely to be extremely ill-conditioned and/or poorly scaled, and the model substructuring process should be revised.
2. Additional kinematic mechanisms are found. If these are disallowed an error exit is taken. If allowed and the spring count agrees with the null space dimension, the geometric **R** is replaced by the updated null space basis and the blocksolver path taken. If acceptable but the spring count disagrees, the factorization path is retraced with adjusted tolerances as above.

The ultimate solution to fatal error conditions is revision of the model decomposition process and imposition of element quality control (in particular, to eliminate the pollution errors discussed in Section 5) until all substructures behave as expected. For this reason, it pays to be conservative in handling error conditions at this level.

A question may be raised as to whether an in-depth null space analysis of the substructural stiffness should be always done on entry. The philosophy of the present implementation is that such analysis serves primarily to catch model or substructuring flaws. Once those are fixed, it is not only superfluous but may contaminate the geometric **R** with roundoff noise. The analysis of complex structural configurations typically requires hundreds or thousands of production runs for design modifications during which the substructuring is kept unchanged. It is better to keep the null space analysis as a background tool for verification and troubleshooting of the initial passes.

A more difficult decision is: should non-RBM substructural kinematic mechanisms be forbidden? Our recommendation is to allow only those that possess physical significance; for example, vehicle steering and suspension linkages, or control surface motions in aircraft. These mechanisms can often be defined geometrically by inspection, as in the example of Fig. 11, again making a detailed null space analysis of **K** superfluous.

8. Conclusions

The introduction of the free–free flexibility closes a duality gap with the stiffness method. This paper treats the subject at the level of individual substructures that arise from a model decomposition process. The major new contributions presented here are:

1. An exact penalty method to compute **F** or a subset thereof, given **K** and **R**, for arbitrary substructures. The method preserves the sparseness structure of **K** and can be implemented in the framework of existing

symmetric sparse linear solvers. To increase robustness it is recommended to complement the basic algorithm with a null space analyzer to handle difficult or borderline cases.
2. A general expression for **F**, derived through the Woodbury formula, that involves a rank-regularization matrix **H**.
3. Congruential transformation methods for symbolic derivation of the free–free flexibility of individual elements.
4. Dual and nondual relations that connect boundary-reduced stiffness and flexibility matrices, which are expected to be useful in model reduction and system identification.

Further refinements in null space analysis of substructures and subdomains can be expected as interest grows in multilevel parallel solution of extremely large FEM models, containing tens or hundreds of millions of freedoms. The solution framework may involve boundary flexibilities connected by displacement frames and should be able to accommodate nonmatching meshes as well as nonlocal (inter-subdomain) multipoint constraints.

Acknowledgements

The present work has been supported by Lawrence Livermore National Laboratory under the Scalable Algorithms for Massively Parallel Computations (ASCI level-II) contract B347880, by Sandia National Laboratory under the Accelerated Strategic Computational Initiative (ASCI) contract AS-5666, and by the National Science Foundation under award ECS-9725504.

Appendix A. Background theory

A.1. Generalized inverse terminology

We summarize below terminology concerning the generalized inverses that are used in the development of Sections 3–5. For a complete coverage of the subject the monograph by Rao and Mitra [21] may be consulted.

A.1.1. Pseudo and spectral g-inverses of nondefective square matrices

Consider a $n \times n$ square real matrix **A**, which may be unsymmetric and singular but is assumed nondefective (that is, it has a complete eigensystem). Its *pseudo-inverse*, also called the Moore–Penrose generalized inverse or *g-inverse*, is the matrix **X** that satisfies the four Penrose conditions

$$\mathbf{AXA} = \mathbf{A}, \quad \mathbf{XAX} = \mathbf{X}, \quad \mathbf{AX} = (\mathbf{AX})^{\mathrm{T}}, \quad \mathbf{XA} = (\mathbf{XA})^{\mathrm{T}}. \tag{A.1}$$

The pseudo-inverse is identified as $\mathbf{X} = \mathbf{A}^{+}$ below. It can be shown that this matrix exists and is unique [10]. An equivalent definition is

$$\mathbf{AX} = \mathbf{P}_A, \quad \mathbf{XA} = \mathbf{P}_X, \tag{A.2}$$

where $\mathbf{P}_A$ and $\mathbf{P}_X$ are projection operators associated with the column space of **A** and **X**, respectively. If **A** is symmetric, so is **X** and $\mathbf{P}_A = \mathbf{P}_X = \mathbf{P}$.

The spectral generalized inverse of **A**, or simply *sg-inverse*, is the matrix $\mathbf{A}^{\dagger}$ that has the same eigenvectors as **A**, and whose nonzero eigenvalues are the reciprocals of the corresponding nonzero eigenvalues of **A**. (The name and notation for this class of g-inverses is not standardized in the literature; see [21, Sec 4.7] for other identifiers.) More precisely, if the nonzero eigenvalues of **A** are λ_i and associated left and right bi-orthonormalized eigenvectors are $\mathbf{x}_i$ and $\mathbf{y}_i$, respectively, we have

$$\mathbf{A} = \sum_i \lambda_i \mathbf{x}_i \mathbf{y}_i^{\mathrm{T}}, \quad \mathbf{A}^{\dagger} = \sum_i \frac{1}{\lambda_i} \mathbf{x}_i \mathbf{y}_i^{\mathrm{T}}, \quad \lambda_i \neq 0, \quad \mathbf{x}_i^{\mathrm{T}} \mathbf{y}_j = \delta_{ij}. \tag{A.3}$$

where δ_{ij} is the Kronecker delta. If $\mathbf{A}$ is symmetric, $\mathbf{A}^{+} = \mathbf{A}^{\dagger}$. For unsymmetric matrices these two g-inverses generally differ. It can be shown [21] that $\mathbf{A}^{\dagger}$ always satisfies the first two Penrose conditions (A.1) but not necessarily the others. Note, however, that $\mathbf{A}$, $\mathbf{A}^{+}$ and $\mathbf{A}^{\dagger}$ have the same rank.

A.1.2. g-Inverses of rectangular matrices, products and sums

The definition of pseudo-inverse may be extended to rectangular matrices with real or complex elements. The general expressions for rectangular real $\mathbf{A}$ are

$$\mathbf{A}^{+} = (\mathbf{A}^{\mathrm{T}}\mathbf{A})^{+}\mathbf{A}^{\mathrm{T}} = \mathbf{A}^{\mathrm{T}}(\mathbf{A}\mathbf{A}^{\mathrm{T}})^{+}, \quad (\mathbf{A}^{\mathrm{T}})^{+} = (\mathbf{A}\mathbf{A}^{\mathrm{T}})^{+}\mathbf{A} = \mathbf{A}(\mathbf{A}^{\mathrm{T}}\mathbf{A})^{+}. \tag{A.4}$$

This reduces the problem to the pseudo-inversion of a symmetric matrix. (For complex matrices transposes are replaced by conjugate-transposes.) On the other hand, the definition (A.3) of sg-inverse is restricted to square matrices.

Let $\mathbf{A}$ and $\mathbf{B}$ be arbitrary rectangular matrices such that $\mathbf{AB}$ is defined. Let $\mathbf{B}_1 = \mathbf{A}^{+}\mathbf{AB}$ and $\mathbf{A}_1 = \mathbf{AB}_1\mathbf{B}_1^{+}$. Then $\mathbf{AB} = \mathbf{A}_1\mathbf{B}_1$ and $(\mathbf{AB})^{+} = \mathbf{B}_1^{+}\mathbf{A}_1^{+}$ [22]. Of particular interest is the case of symmetric square real $\mathbf{A}$ and rectangular real $\mathbf{B}$ with the following common-projector property: $\mathbf{A}^{+}\mathbf{A} = \mathbf{P}$, $\mathbf{PB} = \mathbf{B}$, $\mathbf{BB}^{+} = \mathbf{P}$. Then $\mathbf{B}_1 = \mathbf{PB} = \mathbf{B}$, $\mathbf{A}_1 = \mathbf{AP} = \mathbf{A}$ and $(\mathbf{AB})^{+} = \mathbf{B}^{+}\mathbf{A}^{+}$. Applying this to the congruential transformation $\mathbf{B}^{\mathrm{T}}\mathbf{AB}$ yields

$$(\mathbf{B}^{\mathrm{T}}\mathbf{AB})^{+} = \mathbf{B}^{+}\mathbf{A}^{+}(\mathbf{B}^{+})^{\mathrm{T}}. \tag{A.5}$$

In particular, $(\mathbf{B}^{\mathrm{T}}\mathbf{AB})^{+} = \mathbf{BA}^{+}\mathbf{B}^{\mathrm{T}}$ if $\mathbf{B}^{\mathrm{T}}\mathbf{B} = \mathbf{I}$, which can be verified directly.

The pseudo-inverse of a sum of real matrices $\mathbf{A} = \sum_i \mathbf{A}_i$ is generally a complicated function of the $\mathbf{A}_i$ [23]. There is, however, a simple orthogonality result [21, p. 67]:

$$\mathbf{A}^{+} = \sum_i \mathbf{A}_i^{+} \quad \text{if } \mathbf{A}_i\mathbf{A}_j^{\mathrm{T}} = \mathbf{0} \text{ and } \mathbf{A}_j^{\mathrm{T}}\mathbf{A}_i = \mathbf{0}, \ i \neq j. \tag{A.6}$$

A.2. The inversion of modified matrices

This section summarizes major results concerning the inversion of a general square matrix modified by a lower rank update. For a more thorough coverage of the subject the reader is referred to the survey articles by Hager [24] or Henderson and Searle [25], and references therein.

A.2.1. The Woodbury formula

Let $\mathbf{A}$ be a square nonsingular $n \times n$ matrix, the inverse of which is available. This will be called the *baseline matrix*. Let $\mathbf{U}$ and $\mathbf{V}$ be two $n \times k$ matrices of rank k, with $1 \leqslant k \leqslant n$, and $\mathbf{S}$ be an invertible $k \times k$ matrix. Assume that a nonsingular $k \times k$ matrix $\mathbf{T}$, which satisfies $\mathbf{T}^{-1} + \mathbf{S}^{-1} = \mathbf{V}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{U}$, exists. The inverse of the baseline matrix $\mathbf{A}$ modified by $\Delta\mathbf{A} = -\mathbf{USV}^{\mathrm{T}}$ is given by the identity

$$\mathbf{A}_{\Delta}^{-1} = (\mathbf{A} + \Delta\mathbf{A})^{-1} = \left(\mathbf{A} - \mathbf{USV}^{\mathrm{T}}\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{UTV}^{\mathrm{T}}\mathbf{A}^{-1}. \tag{A.7}$$

This is known as the Woodbury formula [26]. The notation used here is that of Householder [27, p. 124]. Eq. (A.7) is called the Sherman–Morrison formula [28] if $k = 1$, in which case $\mathbf{U}$ and $\mathbf{V}$ are column vectors and $\mathbf{S}$ and $\mathbf{T}$ reduce to scalars. The historical development of these important formulas is discussed in the aforementioned surveys [24,25].

The Woodbury formula can be verified by direct multiplication. A less known way to prove it makes use of the block-bordered system

$$\begin{bmatrix} \mathbf{A} & \mathbf{U} \\ \mathbf{V}^{\mathrm{T}} & \mathbf{S}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \mathbf{G} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{0} \end{bmatrix}, \tag{A.8}$$

in which $\mathbf{I}_n$ is the $n \times n$ identity matrix. Elimination in the order $\mathbf{G} \to \mathbf{B}$ gives $(\mathbf{A} - \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}})\mathbf{B} = \mathbf{I}_n$ whence $\mathbf{B} = \mathbf{A}_{\Delta}^{-1}$. Elimination in the order $\mathbf{B} \to \mathbf{G}$ and backsubstitution for $\mathbf{B}$ gives (A.7). Note that $\mathbf{G} = \mathbf{T}\mathbf{V}^{\mathrm{T}}\mathbf{A}^{-1}$. Both (A.7) and (A.8) are of interest to derive efficient computational methods for $\mathbf{A}_{\Delta}^{-1}$ when $\mathbf{A}$ is a *sparse matrix* that can be factored without pivoting, e.g., a symmetric positive-definite matrix.

A.2.2. Singular baseline matrix

Suppose next that $\mathbf{A}$ is singular with rank $n - k$, whereas $\mathbf{A}_{\Delta} = \mathbf{A} + \Delta\mathbf{A} = \mathbf{A} - \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}}$ has full rank n. If so the Woodbury formula (A.7) fails, and extensions to cover pseudo-inverses [23,29] are too complicated to be of practical use.

Although (A.8) works in principle for singular $\mathbf{A}$ because the bordered coefficient matrix is nonsingular, partial or full pivoting would be necessary in the forward factorization pass. This hinders the computational efficiency if $\mathbf{A}$ is sparse. It is possible to restore efficiency as follows. The singular $\mathbf{A}$ is rendered nonsingular by adding and subtracting a *diagonal* matrix $\mathbf{D}_{\mathrm{H}} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$, of rank k:

$$\mathbf{A}_{\Delta} = \mathbf{A} + \mathbf{H}\mathbf{H}^{\mathrm{T}} - \mathbf{H}\mathbf{H}^{\mathrm{T}} - \mathbf{U}\mathbf{S}\mathbf{V}^{\mathrm{T}} = \mathbf{A} + \mathbf{H}\mathbf{H}^{\mathrm{T}} - \begin{bmatrix} \mathbf{U} & \mathbf{H} \end{bmatrix} \begin{bmatrix} \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{V} \\ \mathbf{H} \end{bmatrix}^{\mathrm{T}}, \tag{A.9}$$

where $\mathbf{I}_k$ is the $k \times k$ identity matrix. The Woodbury formula is applied with $\mathbf{A} + \mathbf{H}\mathbf{H}^{\mathrm{T}}$ as baseline matrix. The equivalent block-bordered form is

$$\begin{bmatrix} \mathbf{A} + \mathbf{H}\mathbf{H}^{\mathrm{T}} & \mathbf{U} & \mathbf{H} \\ \mathbf{V}^{\mathrm{T}} & \mathbf{S}^{-1} & \mathbf{0} \\ \mathbf{H}^{\mathrm{T}} & \mathbf{0} & \mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{B} \\ \mathbf{G}_U \\ \mathbf{G}_{\mathrm{H}} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_n \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}. \tag{A.10}$$

Because $\mathbf{D}_{\mathrm{H}} = \mathbf{H}\mathbf{H}^{\mathrm{T}}$ is diagonal, it does not affect the sparseness of $\mathbf{A}$. If $\mathbf{A}$ is symmetric and $\mathbf{A} + \mathbf{D}_{\mathrm{H}}$ positive definite, the coefficient matrix in (A.10) may be *stably factored without pivoting*. In fact, it is not even necessary to form $\mathbf{A} + \mathbf{H}\mathbf{H}^{\mathrm{T}}$ explicitly because the diagonal entries of $\mathbf{H}$ can be inserted "on the fly" as explained in Section 7. Upon backsubstitution $\mathbf{A}_{\Delta}^{-1}$ appears in $\mathbf{B}$.

A.3. The inversion of modified symmetric matrices

This final section specializes the results of the previous section to symmetric baseline matrices modified by a lower rank symmetric update. The case of singular baseline matrix, which is relevant to the methods of Section 7, is covered in more detail.

A.3.1. Invertible baseline matrices

If the $n \times n$ baseline matrix $\mathbf{A}$ is symmetric and invertible, and the update is $\Delta\mathbf{A} = +\mathbf{U}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U}$ is $n \times k$ and has rank k, we only need to set $\mathbf{S} = -\mathbf{I}_k$ and $\mathbf{V} = \mathbf{U}$ in (A.8). The Woodbury formula becomes $(\mathbf{A} + \mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{U}\mathbf{T}\mathbf{U}^{\mathrm{T}}\mathbf{A}^{-1}$, where $\mathbf{T}^{-1} = \mathbf{U}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{U} + \mathbf{I}_k$. The explicit inversions can be avoided by solving three linear systems and combining the results:

$$\mathbf{A}\mathbf{X} = \mathbf{I}_n, \quad \mathbf{A}\mathbf{Y} = \mathbf{U}, \quad \left(\mathbf{Y}^{\mathrm{T}}\mathbf{U} + \mathbf{I}_k\right)\mathbf{Z}^{\mathrm{T}} = \mathbf{Y}^{\mathrm{T}}, \quad \mathbf{A}_{\Delta}^{-1} = \mathbf{X} - \mathbf{Y}\mathbf{Z}^{\mathrm{T}}. \tag{A.11}$$

This sequence requires solving for $n + k$ right-hand sides with the $n \times n$ coefficient matrix $\mathbf{A}$, and n right-hand sides with the $k \times k$ coefficient matrix $\mathbf{Y}^{\mathrm{T}}\mathbf{U} + \mathbf{I}_k$, which is symmetric because $\mathbf{Y}^{\mathrm{T}}\mathbf{U} = \mathbf{U}^{\mathrm{T}}\mathbf{A}^{-1}\mathbf{U}$. This has computational advantages if $\mathbf{A}$ is sparse and $k \ll n$.

A.3.2. Null space relations

The scenario relevant to the development of the free–free flexibility theory starts with $\mathbf{A}$ as a symmetric singular matrix of rank $n-k$. Here $\mathbf{A}$ can stand for either $\mathbf{K}$ or $\mathbf{F}$, since all formulas are dual. The k-dimensional null space of $\mathbf{A}$ is spanned by the $n\times k$ orthonormal basis matrix $\mathbf{R}$ satisfying $\mathbf{AR}=\mathbf{0}$ and $\mathbf{R}^{\mathrm{T}}\mathbf{R}=\mathbf{I}_k$. The columns $\mathbf{r}_i$ of $\mathbf{R}$ are the null eigenvectors of $\mathbf{A}$. The other $n-k$ orthonormalized eigenvectors $\mathbf{v}_i$ that span the range space of $\mathbf{A}$ are columns of a matrix $\mathbf{V}$. The nonzero eigenvalues of $\mathbf{A}$ are $\lambda_i=\mathbf{v}_i^{\mathrm{T}}\mathbf{A}\mathbf{v}_i$. We consider symmetric updates $\Delta\mathbf{A}=\mathbf{U}\mathbf{U}^{\mathrm{T}}$ such that $\mathbf{A}+\mathbf{U}\mathbf{U}^{\mathrm{T}}$ has full rank n. Therefore $\mathbf{U}$ must have the spectral decomposition

$$\mathbf{U}=\mathbf{V}\mathbf{W}+\mathbf{R}\mathbf{J}, \tag{A.12}$$

where the $k\times k$ matrix $\mathbf{J}=\mathbf{R}^{\mathrm{T}}\mathbf{U}$ has rank k whereas the $(n-k)\times(n-k)$ matrix $\mathbf{W}$ is arbitrary. A long and involved analysis based on Woodbury's formula applied to $(\mathbf{A}+\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1}=[(\mathbf{A}+\mathbf{R}\mathbf{J}\mathbf{J}^{\mathrm{T}}\mathbf{R}^{\mathrm{T}})+\mathbf{V}\mathbf{W}\mathbf{W}^{\mathrm{T}}\mathbf{V}^{\mathrm{T}}]^{-1}$ reveals that

$$\mathbf{V}^{\mathrm{T}}(\mathbf{A}+\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1}\mathbf{V}=\mathrm{diag}(1/\lambda_i), \tag{A.13}$$

although the $\mathbf{v}_i$ are *not* generally eigenvectors of $(\mathbf{A}+\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1}$ since (A.12) couples the null and range eigenspaces unless $\mathbf{W}=\mathbf{0}$. From (A.13) follows the important property

$$\mathbf{U}^{\mathrm{T}}(\mathbf{A}+\mathbf{U}\mathbf{U}^{\mathrm{T}})^{-1}\mathbf{U}=\mathbf{I}_k. \tag{A.14}$$

The specialization $\mathbf{U}\to\mathbf{R}$ gives the following identities, of which the first two follow easily from spectral analysis, and the last one from either (A.14) or $\mathbf{R}^{\mathrm{T}}\mathbf{R}=\mathbf{I}_k$:

$$\mathbf{R}^{\mathrm{T}}(\mathbf{A}+\mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}=\mathbf{R}^{\mathrm{T}},\quad (\mathbf{A}+\mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{R}=\mathbf{R},\quad \mathbf{R}^{\mathrm{T}}(\mathbf{A}+\mathbf{R}\mathbf{R}^{\mathrm{T}})^{-1}\mathbf{R}=\mathbf{I}_k. \tag{A.15}$$

A second useful specialization is $\mathbf{U}\to\mathbf{H}$, the rank regularization matrix introduced in the previous section. From $\mathbf{H}^{\mathrm{T}}(\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{H}=\mathbf{H}^{\mathrm{T}}\mathbf{X}\mathbf{H}=\mathbf{I}_k$ and (A.15) it is not difficult to find the following relations, which hold for any $\mathbf{H}$ as long as $\mathrm{rank}(\mathbf{J})=k$. For notational brevity introduce $\mathbf{Y}_{\mathrm{H}}=(\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{H}=\mathbf{X}\mathbf{H}$ and $\mathbf{Y}_{\mathrm{R}}=(\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}})^{-1}\mathbf{R}=\mathbf{X}\mathbf{R}$. Then

$$\begin{aligned}&\mathbf{H}^{\mathrm{T}}\mathbf{Y}_{\mathrm{H}}=\mathbf{I}_k,\quad \mathbf{Y}_{\mathrm{H}}^{\mathrm{T}}\mathbf{R}=\mathbf{H}^{\mathrm{T}}\mathbf{Y}_{\mathrm{R}},\quad \mathbf{Y}_{\mathrm{H}}\mathbf{H}^{\mathrm{T}}\mathbf{R}=\mathbf{R},\quad \mathbf{Y}_{\mathrm{H}}^{\mathrm{T}}\mathbf{R}\mathbf{R}^{\mathrm{T}}=\mathbf{Y}_{\mathrm{H}}^{\mathrm{T}},\\ &\mathbf{H}^{\mathrm{T}}\mathbf{Y}_{\mathrm{R}}\mathbf{R}^{\mathrm{T}}=\mathbf{Y}_{\mathrm{H}}^{\mathrm{T}},\quad \mathbf{Y}_{\mathrm{H}}^{\mathrm{T}}\mathbf{Y}_{\mathrm{H}}=(\mathbf{H}^{\mathrm{T}}\mathbf{R}\mathbf{R}^{\mathrm{T}}\mathbf{H})^{-1},\quad \mathbf{Y}_{\mathrm{H}}(\mathbf{Y}_{\mathrm{H}}^{\mathrm{T}}\mathbf{Y}_{\mathrm{H}})^{-1}\mathbf{Y}_{\mathrm{H}}^{\mathrm{T}}=\mathbf{R}\mathbf{R}^{\mathrm{T}}.\end{aligned} \tag{A.16}$$

A.3.3. Computing $\mathbf{B}$ *with linear solvers*

We are now ready to tackle the efficient computation of $\mathbf{B}$ in

$$(\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}}-\mathbf{H}\mathbf{H}^{\mathrm{T}}+\mathbf{R}\mathbf{R}^{\mathrm{T}})\mathbf{B}=\mathbf{P}, \tag{A.17}$$

where $\mathbf{P}=\mathbf{I}-\mathbf{R}\mathbf{R}^{\mathrm{T}}$. If $\mathbf{A}$ is $\mathbf{K}$ or $\mathbf{F}$, $\mathbf{B}$ gives $\mathbf{F}$ or $\mathbf{K}$, respectively; the first case being more important in practice. Using $\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}}$ as baseline matrix, the equivalent block bordered system is

$$\begin{bmatrix}\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}} & \mathbf{R} & \mathbf{H}\\ \mathbf{R}^{\mathrm{T}} & -\mathbf{I}_k & \mathbf{0}\\ \mathbf{H}^{\mathrm{T}} & \mathbf{0} & \mathbf{I}_k\end{bmatrix}\begin{bmatrix}\mathbf{B}\\ \mathbf{G}_{\mathrm{R}}\\ \mathbf{G}_{\mathrm{H}}\end{bmatrix}=\begin{bmatrix}\mathbf{P}\\ \mathbf{0}\\ \mathbf{0}\end{bmatrix}. \tag{A.18}$$

Applying the staged solving sequence (A.11) while accounting for the presence of $\mathbf{P}$ on the right gives

$$(\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}})\mathbf{X}=\mathbf{I},\quad (\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}})\mathbf{Y}_{\mathrm{R}}=\mathbf{R},\quad (\mathbf{A}+\mathbf{H}\mathbf{H}^{\mathrm{T}})\mathbf{Y}_{\mathrm{H}}=\mathbf{H}, \tag{A.19}$$

$$\begin{bmatrix} \mathbf{Y}_R^T\mathbf{R}+\mathbf{I}_k & \mathbf{Y}_R^T\mathbf{H} \\ \mathbf{Y}_H^T\mathbf{R} & \mathbf{Y}_H^T\mathbf{H}-\mathbf{I}_k \end{bmatrix} \begin{bmatrix} \mathbf{Z}_R^T \\ \mathbf{Z}_H^T \end{bmatrix} = \begin{bmatrix} \mathbf{Y}_R^T \\ \mathbf{Y}_H^T \end{bmatrix}, \tag{A.20}$$

$$\mathbf{B} = \big(\mathbf{X} - \mathbf{Y}_R\mathbf{Z}_R^T - \mathbf{Y}_H\mathbf{Z}_H^T\big)\mathbf{P}. \tag{A.21}$$

This sequence may be considerably simplified by taking advantage of the relation catalog (A.16). Since $\mathbf{Y}_H^T\mathbf{H} = \mathbf{I}_k$ the lower diagonal block of the coefficient matrix of (A.20) vanishes. The second matrix equation becomes $\mathbf{Y}_H^T\mathbf{R}\mathbf{Z}_R^T = \mathbf{Y}_H^T$. Comparing this to the fourth of (A.16) shows that $\mathbf{Z}_R^T = \mathbf{R}^T$, whence the term $\mathbf{Y}_R\mathbf{Z}_R^T\mathbf{P}$ in (A.21) drops out since $\mathbf{R}^T\mathbf{P} = \mathbf{0}$. Pre-multiplying the first matrix equation of (A.20) by $\mathbf{Y}_H^T$ and solving for the remaining unknown gives $\mathbf{Z}_H^T = \mathbf{H}\mathbf{R}\mathbf{R}^T\mathbf{H}\mathbf{Y}_H^T(\mathbf{R}\mathbf{Y}_R - \mathbf{I}) = \mathbf{H}^T\mathbf{R}(\mathbf{Y}_R^T - \mathbf{R}^T)$. Substitution into (A.21) and use of the last of (A.16) yields

$$\mathbf{B} = \mathbf{P}\mathbf{X}\mathbf{P} = \mathbf{P}(\mathbf{A}+\mathbf{H}\mathbf{H}^T)^{-1}\mathbf{P} = \mathbf{X} - \mathbf{Y}_R\mathbf{R}^T - \mathbf{R}\mathbf{Y}_R^T + \mathbf{R}\mathbf{R}^T\mathbf{Y}_R\mathbf{R}^T. \tag{A.22}$$

This shows that the solution $\mathbf{Y}_H$ of the third linear system in (A.19) is not actually needed. In fact, the only place where $\mathbf{H}$ appears is in the regularization of $\mathbf{A}$ to $\mathbf{A}_H$.

The final result (A.22) holds for arbitrary $\mathbf{H}$ such that $\mathbf{J} = \mathbf{H}^T\mathbf{R}$ has rank k. (It is not restricted to diagonal $\mathbf{H}\mathbf{H}^T$, which is used in the actual computations because it preserves sparsity.) In particular, if $\mathbf{H} = \mathbf{R}$ we recover the third forms listed in (16) and (17). In this special case one of the $\mathbf{P}$ multipliers may be dropped because of (A.15). But for general $\mathbf{H}$ the result is at first sight puzzling. The spectrum of $\mathbf{X} = (\mathbf{A}+\mathbf{H}\mathbf{H}^T)^{-1}$ does not look at all like that of $\mathbf{A}$ because the null and range eigenspaces $\mathbf{R}$ and $\mathbf{V}$ of $\mathbf{A}$ become strongly entangled via $\mathbf{H}$, and zero eigenvalues disappear. So how come $\mathbf{B} = \mathbf{P}\mathbf{X}\mathbf{P}$?

The answer to the puzzle is (A.13): the range eigenspace $\mathbf{V}$ is imprinted in the spectral morass of $\mathbf{X}$. Two projector applications disentangle $\mathbf{V}$ and restore the null space.

References

[1] C. Farhat, F.-X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, Int. J. Numer. Methods Engrg. 32 (1991) 1205–1227.

[2] C. Farhat, F.-X. Roux, Implicit parallel processing in structural mechanics, Comput. Mech. Adv. 2 (1994) 1–124.

[3] K.C. Park, M.R. Justino F, C.A. Felippa, An algebraically partitioned FETI method for parallel structural analysis: algorithm description, Int. J. Numer. Methods Engrg. 40 (1997) 2717–2737.

[4] M.R. Justino F, K.C. Park, C.A. Felippa, An algebraically partitioned FETI method for parallel structural analysis: performance evaluation, Int. J. Numer. Methods Engrg. 40 (1997) 2739–2758.

[5] U. Gumaste, K.C. Park, K.F. Alvin, A family of implicit partitioned time integration algorithms for parallel analysis of heterogeneous structural systems, Comput. Mech. 24 (2000) 463–475.

[6] K.C. Park, K.F. Alvin, Extraction of substructural flexibility from measured global modes and mode shapes, in: AIAA 96-1297, Proceedings of the 1996 AIAA SDM Conference, Salt Lake City, Utah, April 1996, AIAA J. 35 (1997) 1187–1194.

[7] K.C. Park, G.W. Reich, K.F. Alvin, Damage detection using localized flexibilities, in: F.-K. Chang (Ed.), Structural Health Monitoring, Current Status and Perspectives, Technomic Publishers, 1997, pp. 125–139.

[8] K.C. Park, G.W. Reich, A theory for strain-based structural system identification, in: Proceedings of the 9th International Conference on Adaptive Structures and Technologies, Cambridge, MA, October 14–16, 1998.

[9] K.C. Park, C.A. Felippa, A flexibility-based inverse algorithm for identification of structural joint properties, in: Proceedings of the ASME Symposium on Computational Methods on Inverse Problems, Anaheim, CA, November 15–20, 1998.

[10] G.W. Stewart, J. Sun, Matrix Perturbation Theory, Academic Press, New York, 1990.

[11] C.A. Felippa, K.C. Park, M.R. Justino F, The construction of free–free flexibility matrices as generalized stiffness inverses, Comput. Struct. 88 (1998) 411–418.

[12] C.A. Felippa, K.C. Park, A direct flexibility method, Comput. Methods Appl. Mech. Engrg. 149 (1997) 319–337.

[13] S.L. Bott, R.J. Duffin, On the algebra of networks, Trans. Am. Math. Soc. 74 (1953) 99–109.

[14] C.A. Felippa, Recent advances in finite element templates, in: B.H.V. Topping (Ed.), Computational Mechanics for the Twenty-First Century, Saxe-Coburn Publications, Edinburgh, 2000, pp. 71–98 (Chapter 4).

[15] C.A. Felippa, Penalty spring stabilization of singular Jacobians, J. Appl. Mech. 54 (1987) 1730–1733.
[16] C.A. Felippa, Traversing critical points by penalty springs, in: Proceedings of NUMETA'87 Conference, Swansea, Wales, July, M. Nijhoff, Dordrecht, Netherlands, 1987.
[17] C.A. Felippa, Solution of equations with skyline–stored symmetric coefficient matrix, Comput. Struct. 5 (1975) 13–25.
[18] C.L. Lawson, R.J. Hanson, Solving Least Squares Problems, Prentice-Hall, Englewood Cliffs, NJ, 1974.
[19] C. Farhat, M. Geradin, On the general solution by a direct method of a large scale singular system of equations: application to the analysis of floating structures, Int. J. Numer. Methods Engrg. 41 (1998) 675–696.
[20] M. Papadrakakis, Y. Fragakis, An integrated geometric–algebraic approach for solving semi-definite problems in structural mechanics, Comput. Methods Appl. Mech. Engrg. 190 (2001) 6513–6532.
[21] C.R. Rao, S.K. Mitra, Generalized Inverse of Matrices and its Applications, Wiley, New York, 1971.
[22] R.E. Cline, Note on the generalized inverse of the product of matrices, SIAM Rev. 6 (1964) 57–58.
[23] S.L. Campbell, C.D. Meyer, Generalized Inverses of Linear Transformations, Dover, New York, 1991.
[24] W.W. Hager, Updating the inverse of a matrix, SIAM Rev. 31 (1989) 221–239.
[25] H.V. Henderson, S.R. Searle, On deriving the inverse of a sum of matrices, SIAM Rev. 23 (1981) 53–60.
[26] M. Woodbury, Inverting modified matrices, Memorandum Report 42, Statistical Research Group, Princeton University, Princeton, NJ, 1950.
[27] A. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell, New York, 1964.
[28] J. Sherman, W.J. Morrison, Adjustment of an inverse matrix corresponding to changes in the elements of a given column or a given row of the original matrix, Ann. Math. Statist. 20 (1949) 621; Also, Adjustments of an inverse matrix corresponding to a change in one element of a given matrix, Ann. Math. Statist. 21 (1950) 124.
[29] J.A. Fill, D.E. Fishkind, The Moore–Penrose generalized inverse for sum of matrices, SIAM J. Matrix Anal. Appl. 21 (1999) 629–635.