10

Superelements and Global-Local Analysis

TABLE OF CONTENTS

§10.1. Superelement Concept

A superelement is a grouping of finite elements that, upon assembly, may be regarded as an *individual element* for computational purposes. These purposes may be driven by modeling or processing needs.

A random assortment of elements does not necessarily make up a superelement. To be considered as such, a grouping must meet certain conditions. Informally we can say that it must form a structural component on its own. This imposes certain conditions stated mathematically in §10.1.3. Inasmuch as these conditions involve advanced concepts such as rank sufficiency, which are introduced in later Chapters, the restrictions are not dwelled upon here.

As noted in Chapter 6, superelements may originate from two overlapping contexts: "bottom up" or "top down." In a bottom up context one thinks of superelements as built from simpler elements. In a top-down context, superelements may be thought as being large pieces of a complete structure. This dual viewpoint motivates the following classification:

Macroelements. These are superelements assembled with a few primitive elements. Also called *mesh units* when they are presented to program users as individual elements.

Substructures. Complex assemblies of elements that result on breaking up a structure into distinguishable portions.

When does a substructure becomes a macroelement or vice-versa? There are no precise rules. In fact the generic term *superelement* was coined to cover the entire spectrum, ranging from *individual elements* to *complete structures*. This universality is helped by common processing features.

Both macroelements and substructures are treated exactly the same way as regards matrix processing. The basic rule is that associated with *condensation* of internal degrees of freedom. The technique is illustrated in the following section with a simple example. The reader should note, however, that condensation applies to *any* superelement, whether composed of two or a million elements.[1]

§10.1.1. Where Does the Idea Come From?

Substructuring was invented by aerospace engineers in the early 1960s[2] to carry out a first-level breakdown of complex systems such as a complete airplane, as depicted in Figure 10.1. The decomposition may continue hierarchically through additional levels as illustrated in Figure 10.2. The concept is also natural for space vehicles operating in stages, such as the Apollo short stack depicted in Figure 10.3.

Three original motivating factors for substructuring can be cited.

1. **Facilitate division of labor**. Substructures with different functions are done by separate design groups with specialized knowledge and experience. For instance an aircraft company may set up a fuselage group, a wing group, a landing-gear group, etc. These groups are thus protected from "hurry up and wait" constraints. More specifically: a wing design group can

---

[1] Of course the computer implementation becomes totally different as one goes from macroelements to substructures, because efficient processing for large matrix systems requires exploitation of sparsity.

[2] See **Notes and Bibliography** at the end of this Chapter.

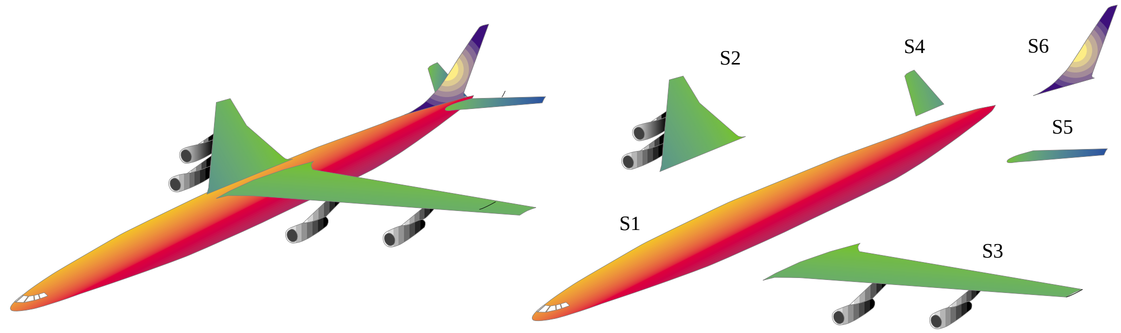

FIGURE 10.1. Complete airplane broken down into six level one substructures identified as S_1 through S_6.

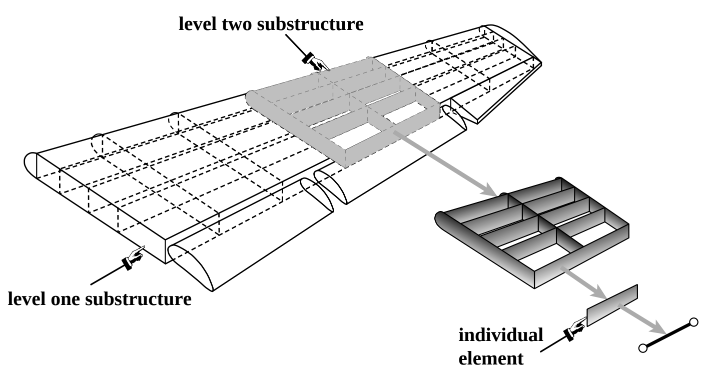

FIGURE 10.2. Further breakdown of wing structure. The decomposition process may continue down to the individual element level.

keep on working on refinements, improvements and experimental model verification as long as the interface information (the wing-fuselage intersection) stays sensibly unchanged.

2. **Take advantage of repetition**. Often structures are built of several identical or nearly identical units. For instance, the wing substructures S_2 and S_3 of Figure 10.1 are mirror images on reflection about the fuselage midplane, and so are the stabilizers S_4 and S_5. Even if the loading is not symmetric about that midplane, recognizing repetitions saves model preparation time.

3. **Overcome computer limitations**. The computers of the 1960s operated under serious memory limitations. (For example, the first supercomputer: the Control Data 6600, had a total high-speed memory of 131072 60-bit words or 1.31 MB; that machine cost \$10M in 1966 dollars.) It was difficult to fit a complex structure such as an airplane as one entity. Substructuring permitted the complete analysis to be carried out in stages through use of auxiliary storage devices such as tapes or disks.

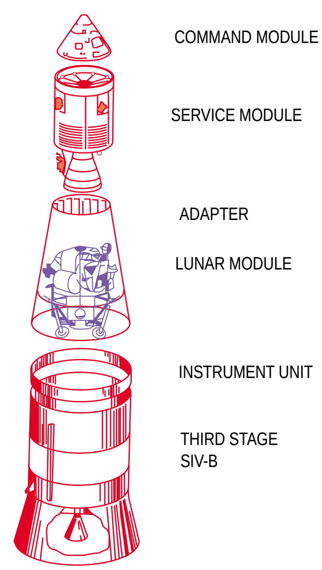

FIGURE 10.3. The Apollo short stack.

Of the three motivations, the first two still hold today. The third one has moved to a different plane:

parallel processing, as noted in §10.1.2 below.

In the late 1960s the idea was picked up and developed extensively by the offshore and shipbuilding industries, the products of which tend to be modular and repetitive to reduce fabrication costs. As noted above, repetition favors the use of substructuring techniques.

At the other end of the superelement spectrum, the mesh units herein called macroelements appeared in the mid 1960s. They were motivated by user convenience. For example, in hand preparation of models, quadrilateral and bricks involve less human labor than triangles and tetrahedra, respectively. It was therefore natural to combine the latter to assemble the former. Going a step further one can assemble components such as "box elements" for applications such as box-girder bridges.

§10.1.2. Subdomains

Applied mathematicians working on solution procedures for parallel computation have developed the concept of *subdomains*. These are groupings of finite elements that are entirely motivated by computational considerations. They are subdivisions of the finite element model done more or less automatically by a program called *domain decomposer*.

Although the concepts of substructures and subdomains overlap in many respects, it is better to keep the two separate. The common underlying theme is divide and conquer but the motivation is different.

§10.1.3. *Mathematical Requirements

A superelement is said to be *rank-sufficient* if its only zero-energy modes are rigid-body modes. Equivalently, the superelement does not possess spurious kinematic mechanisms.

Verification of the rank-sufficient condition guarantees that the static condensation procedure described below will work properly.

§10.2. Static Condensation

Degrees of freedom of a superelement are classified into two groups:

Internal Freedoms. Those that are not connected to the freedoms of another superelement. Nodes whose freedoms are internal are called *internal nodes*.

Boundary Freedoms. These are connected to at least another superelement. They usually reside at *boundary nodes* placed on the periphery of the superelement. See Figure 10.4.

The objective is to get rid of all displacement degrees of freedom associated with *internal freedoms*. This elimination process is called *static condensation*, or simply *condensation*.

Condensation may be presented in terms of explicit matrix operations, as shown in the next subsection. A more practical technique based on symmetric Gauss elimination is discussed later.

§10.2.1. Condensation by Explicit Matrix Operations

To carry out the condensation process, the assembled stiffness equations of the superelement are partitioned as follows:

$$\begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{bmatrix}. \tag{10.1}$$

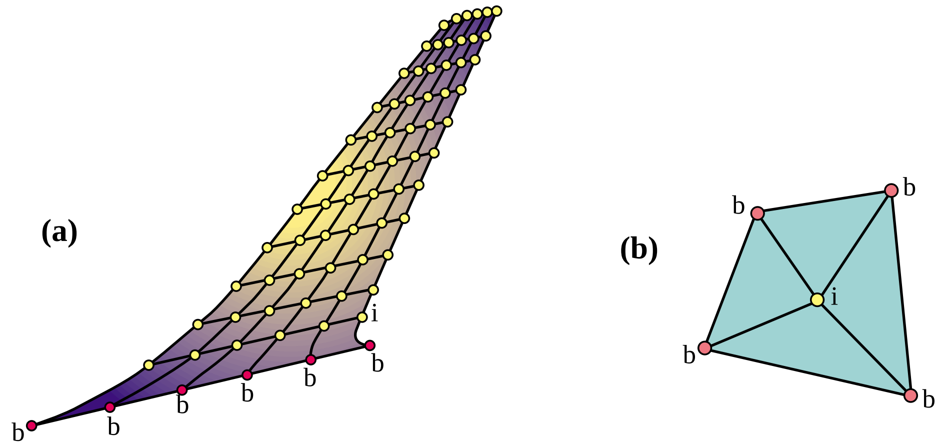

FIGURE 10.4. Classification of superelement freedoms into boundary and internal. (a) shows the vertical stabilizer substructure S_6 of Figure 10.2. (The FE mesh is pictured as two-dimensional for illustrative purposes; for an actual aircraft it will be three dimensional.) Boundary freedoms are those associated to the boundary nodes labeled b (shown in red), which are connected to the fuselage substructure. (b) shows a quadrilateral macroelement mesh-unit fabricated with 4 triangles: it has one interior and four boundary nodes.

where subvectors $\mathbf{u}_b$ and $\mathbf{u}_i$ collect *boundary* and *interior* degrees of freedom, respectively. Take the second matrix equation:

$$\mathbf{K}_{ib}\mathbf{u}_b + \mathbf{K}_{ii}\mathbf{u}_i = \mathbf{f}_i, \tag{10.2}$$

If $\mathbf{K}_{ii}$ is nonsingular we can solve for the interior freedoms:

$$\mathbf{u}_i = \mathbf{K}_{ii}^{-1}(\mathbf{f}_i - \mathbf{K}_{ib}\mathbf{u}_b), \tag{10.3}$$

Replacing into the first matrix equation of (10.1) yields the *condensed stiffness equations*

$$\tilde{\mathbf{K}}_{bb}\mathbf{u}_b = \tilde{\mathbf{f}}_b. \tag{10.4}$$

In this equation,

$$\tilde{\mathbf{K}}_{bb} = \mathbf{K}_{bb} - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib}, \qquad \tilde{\mathbf{f}}_b = \mathbf{f}_b - \mathbf{K}_{bi}\mathbf{K}_{ii}^{-1}\mathbf{f}_i, \tag{10.5}$$

are called the *condensed* stiffness matrix and force vector, respectively, of the substructure.

From this point onward, the condensed superelement may be viewed, from the standpoint of further operations, as an *individual element* whose element stiffness matrix and nodal force vector are $\tilde{\mathbf{K}}_{bb}$ and $\tilde{\mathbf{f}}_b$, respectively.

Often each superelement has its own "local" coordinate system. A transformation of (10.5) to an overall global coordinate system is necessary upon condensation. In the case of multiple levels, the transformation is done with respect to the next-level superelement coordinate system. This coordinate transformation procedure automates the processing of repeated portions.

Remark 10.1. The feasibility of the condensation process (10.3)–(10.5) rests on the non-singularity of $\mathbf{K}_{ii}$. This matrix is nonsingular if the superelement is rank-sufficient in the sense stated in §10.1.3, and if fixing the boundary freedoms precludes all rigid body motions. If the former condition is verified but not the latter, the superelement is called *floating*. Processing floating superelements demands more advanced computational techniques, among which one may cite the use of projectors and generalized inverses [**?**].

```
CondenseLastFreedom[K_,f_]:=Module[{pivot,c,Kc,fc,
    n=Length[K]}, If [n<=1,Return[{K,f}]];
    Kc=Table[0,{n-1},{n-1}]; fc=Table[0,{n-1}];
    pivot=K[[n,n]]; If [pivot==0, Print["CondenseLastFreedom:",
          " Singular Matrix"]; Return[{K,f}]];
    For [i=1,i<=n-1,i++,  c=K[[i,n]]/pivot;
         fc[[i]]=f[[i]]-c*f[[n]];
         For [j=1,j<=i,j++,
             Kc[[j,i]]=Kc[[i,j]]=K[[i,j]]-c*K[[n,j]]
             ];
        ];
    Return[Simplify[{Kc,fc}]]
    ];
```

```
K={{6,-2,-1,-3},{ -2,5,-2,-1},{ -1,-2,7,-4},{-3,-1,-4,8}};
f={3,6,4,0};
Print["Before condensation:"," K=",K//MatrixForm," f=",f//MatrixForm];
{K,f}=CondenseLastFreedom[K,f];Print["Upon condensing DOF 4:",
  " K=",K//MatrixForm," f=",f//MatrixForm];
{K,f}=CondenseLastFreedom[K,f];Print["Upon condensing DOF 3:",
  " K=",K//MatrixForm," f=",f//MatrixForm];
```

FIGURE 10.5. *Mathematica* module to condense the last degree of freedom from a stiffness matrix and force vector. The test statements carry out the example (10.6)–(10.10).

§10.2.2. Condensation by Symmetric Gauss Elimination

In the computer implementation of the the static condensation process, calculations are not carried out as outlined above. There are two major differences. The equations of the substructure are not actually rearranged, and the explicit calculation of the inverse of $\mathbf{K}_{ii}$ is avoided. The procedure may be in fact coded as a variant of symmetric Gauss elimination. To convey the flavor of this technique, consider the following stiffness equations of a superelement:

$$\begin{bmatrix} 6 & -2 & -1 & -3 \\ -2 & 5 & -2 & -1 \\ -1 & -2 & 7 & -4 \\ -3 & -1 & -4 & 8 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 4 \\ 0 \end{bmatrix}. \tag{10.6}$$

Suppose that the last two displacement freedoms: u_3 and u_4, are classified as interior and are to be statically condensed out. To eliminate u_4, perform symmetric Gauss elimination of the fourth row and column:

$$\begin{bmatrix} 6-\frac{(-3)\times(-3)}{8} & -2-\frac{(-1)\times(-3)}{8} & -1-\frac{(-4)\times(-3)}{8} \\ -2-\frac{(-3)\times(-1)}{8} & 5-\frac{(-1)\times(-1)}{8} & -2-\frac{(-4)\times(-1)}{8} \\ -1-\frac{(-3)\times(-4)}{8} & -2-\frac{(-1)\times(-4)}{8} & 7-\frac{(-4)\times(-4)}{8} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 3-\frac{0\times(-3)}{8} \\ 6-\frac{0\times(-1)}{8} \\ 4-\frac{0\times(-4)}{8} \end{bmatrix}, \tag{10.7}$$

or

$$\begin{bmatrix} \frac{39}{8} & -\frac{19}{8} & -\frac{5}{2} \\ -\frac{19}{8} & \frac{39}{8} & -\frac{5}{2} \\ -\frac{5}{2} & -\frac{5}{2} & 5 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 6 \\ 4 \end{bmatrix}. \tag{10.8}$$

Repeat the process for the third row and column to eliminate u_3:

$$\begin{bmatrix} \frac{39}{8}-\frac{(-5/2)\times(-5/2)}{5} & -\frac{19}{8}-\frac{(-5/2)\times(-5/2)}{5} \\ -\frac{19}{8}-\frac{(-5/2)\times(-5/2)}{5} & \frac{39}{8}-\frac{(-5/2)\times(-5/2)}{5} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 3-\frac{4\times(-5/2)}{5} \\ 6-\frac{4\times(-5/2)}{5} \end{bmatrix}, \tag{10.9}$$

or

$$\begin{bmatrix} \frac{29}{8} & -\frac{29}{8} \\ -\frac{29}{8} & \frac{29}{8} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 8 \end{bmatrix}. \tag{10.10}$$

These are the condensed stiffness equations. Figure 10.5 shows a *Mathematica* program that carries out the foregoing steps. Module `CondenseLastFreedom` condenses the last freedom of a stiffness matrix `K` and a force vector `f`. It is invoked as `{ Kc,fc }=CondenseLastFreedom[K,f].` It returns the condensed stiffness `Kc` and force vector `fc` as new arrays. To do the example (10.6)–(10.10), the module is called twice, as illustrated in the test statements of Figure 10.5.

Obviously this procedure is much simpler than going through the explicit matrix inverse. Another important advantage of Gauss elimination is that equation rearrangement is not required even if the condensed degrees of freedom do not appear sequentially. For example, suppose that the assembled superelement contains originally eight degrees of freedom and that the freedoms to be condensed out are numbered 1, 4, 5, 6 and 8. Then Gauss elimination is carried out over those equations only, and the condensed (3×3) stiffness and (3×1) force vector extracted from rows and columns 2, 3 and 7. An implementation of this process is considered in Exercise 10.2.

Remark 10.2. The symmetric Gauss elimination procedure, as illustrated in steps (10.6)–(10.10), is primarily useful for macroelements and mesh units, since the number of stiffness equations for those typically does not exceed a few hundreds. This permits the use of full matrix storage. For substructures containing thousands or millions of degrees of freedom — such as in the airplane example — the elimination is carried out using more sophisticated sparse matrix algorithms; for example that described in [211].

Remark 10.3. The static condensation process is a matrix operation called "partial inversion" or "partial elimination" that appears in many disciplines. Here is the general form. Suppose the linear system $\mathbf{A}\mathbf{x} = \mathbf{y}$, where $\mathbf{A}$ is $n \times n$ square and $\mathbf{x}$ and $\mathbf{y}$ are n-vectors, is partitioned as

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{bmatrix}. \tag{10.11}$$

Assuming the appropriate inverses to exist, then the following are easily verified matrix identities:

$$\begin{bmatrix} \mathbf{A}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \\ \mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_2 \end{bmatrix}, \quad \begin{bmatrix} \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}\mathbf{A}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{A}_{22}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{x}_2 \end{bmatrix}. \tag{10.12}$$

We say that $\mathbf{x}_1$ has been eliminated or "condensed out" in the left identity and $\mathbf{x}_2$ in the right one. In FEM applications, it is conventional to condense out the bottom vector $\mathbf{x}_2$, so the right identity is relevant. If $\mathbf{A}$ is symmetric, to retain symmetry in (10.12) it is necessary to change the sign of one of the subvectors.

§10.2.3. Recovery of Internal Freedoms

After the boundary freedoms are obtained by the solution process, the interior freedoms can be recovered directly from (10.3). This process may be carried out either directly through a sequence of matrix operations, or equation by equation as a backsubstitution process.

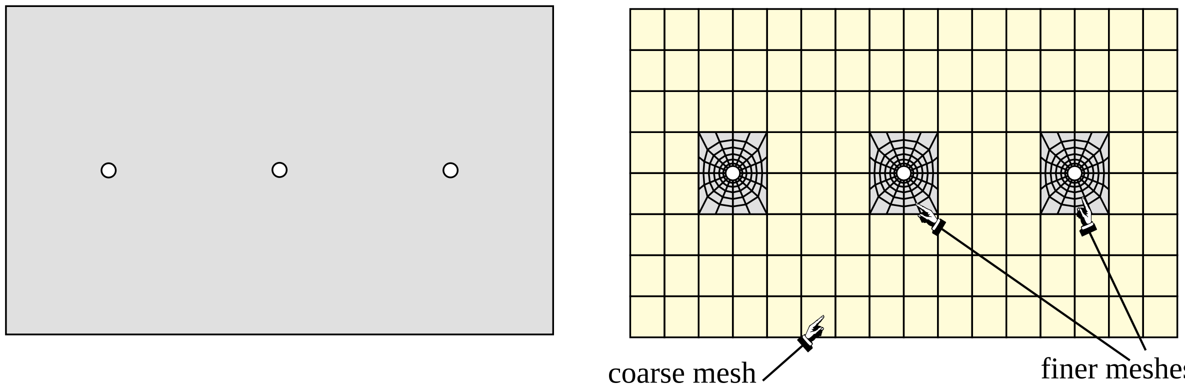

FIGURE 10.6. Left: example panel structure for global-local analysis. Right: a FEM mesh for a one-shot analysis.

§10.3. Global-Local Analysis

As discussed in the first Chapter, complex engineering systems are often modeled in a *multilevel* fashion following the divide and conquer approach. The superelement technique is a practical realization of that approach.

A related, but not identical, technique is *multiscale* analysis. The whole system is first analyzed as a global entity, discarding or passing over details deemed not to affect its overall behavior. Local details are then analyzed using the results of the global analysis as boundary conditions. The process can be continued into the analysis of further details of local models. And so on. When this procedure is restricted to two stages and applied in the context of finite element analysis, it is called *global-local* analysis in the FEM literature.

In the global stage the behavior of the entire structure is simulated with a finite element model that necessarily ignores details such as cutouts or joints. These details do not affect the overall behavior of the structure, but may have a bearing on safety. Such details are *a posteriori* incorporated in a series of local analyses.

The gist of the global-local approach is explained in the example illustrated in Figures 10.6 and 10.7. Although the structure is admittedly too simple to merit the application of global-local analysis, it serves to illustrate the basic ideas. Suppose one is faced with the analysis of the rectangular panel shown on the top of Figure 10.6, which contains three small holes. The bottom of that figure shows a standard (one-stage) FEM treatment using a largely regular mesh that is refined near the holes. Connecting the coarse and fine meshes usually involves using multifreedom constraints because the nodes at mesh boundaries do not match, as depicted in that figure.

Figure 10.7 illustrates the global-local analysis procedure. The global analysis is done with a coarse but regular FEM mesh that *ignores the effect of the holes*. This is followed by local analysis of the region near the holes using refined finite element meshes. The key ingredient for the local analyses is the application of boundary conditions (BCs) on the finer mesh boundaries. These BCs may be of displacement (essential) or of force (natural) type. If the former, the applied boundary displacements are interpolated from the global mesh solution. If the latter, the internal forces or stresses obtained from the global calculation are converted to nodal forces on the fine meshes

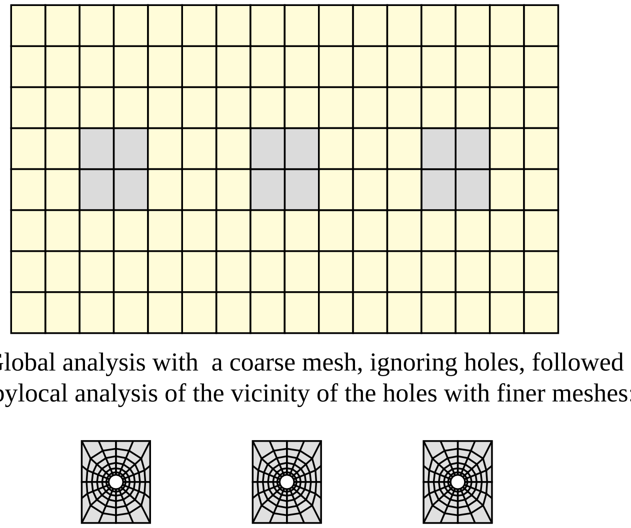

FIGURE 10.7. Global-local analysis of problem of Figure 10.6.

through a lumping process.

The BC choice noted above gives rise to two basic variations of the global-local approach. Experience accumulated over several decades[3] has shown that the stress-BC approach generally gives more reliable answers.

The global-local technique can be extended to more than two levels, in which case it receives the more encompassing name *multiscale analysis*. Although this generalization is still largely in the realm of research, it is receiving increasing attention from various science and engineering communities for complex products such as the thermomechanical analysis of microelectronic components.

Notes and Bibliography

Substructuring was invented by aerospace engineers in the early 1960s. Przemieniecki's book [603] contains a fairly complete bibliography of early work. Most of this was in the form of largely inaccessible internal company or lab reports and so the actual history is difficult to trace. Macroelements appeared simultaneously in many of the early FEM codes. Quadrilateral macroelements fabricated with triangles are described in [205]. For a survey of uses of the static condensation algorithm in FEM, see [809].

The generic term *superelement* was coined in the late 1960s by the SESAM group at DNV Veritas [196]. The matrix form of static condensation for a complete structure is presented in [23], as a scheme to eliminate unloaded DOF in the displacement method. It is unclear when this idea was first applied to substructures or macroelements. The first application for reduced dynamical models is by Guyan [326].

The application of domain decomposition to parallel FEM solvers has produced an enormous and highly specialized literature. Procedures for handling floating superelements using generalized inverse methods are discussed in [234,242,571].

The global-local analysis procedure described here is also primarily used in industry and as such it is rarely mentioned in academic textbooks.

References

Referenced items have been moved to Appendix R.

[3] Particularly in the aerospace industry, in which the global-local technique has been used since the early 1960s.

Homework Exercises for Chapter 10

Superelements and Global-Local Analysis

EXERCISE 10.1 [N:15] The free-free stiffness equations of a superelement are

$$\begin{bmatrix} 88 & -44 & -44 & 0 \\ -44 & 132 & -44 & -44 \\ -44 & -44 & 176 & -44 \\ 0 & -44 & -44 & 220 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 10 \\ 15 \\ 20 \end{bmatrix}. \tag{E10.1}$$

Eliminate u_2 and u_3 from (E10.1) by static condensation, and show (but do not solve) the condensed equation system. Use either the explicit matrix inverse formulas 10.5 or the symmetric Gauss elimination process explained in §10.2.2. Hint: regardless of method the result should be

$$\begin{bmatrix} 52 & -36 \\ -36 & 184 \end{bmatrix} \begin{bmatrix} u_1 \\ u_4 \end{bmatrix} = \begin{bmatrix} 15 \\ 30 \end{bmatrix}. \tag{E10.2}$$

EXERCISE 10.2 [C+N:20] Generalize the program of Figure 10.5 to a module `CondenseFreedom[K,f,k]` that is able to condense the `k`th degree of freedom from `K` and `f`, which is not necessarily the last one. That is, `k` may range from 1 to `n`, where `n` is the number of freedoms in $\mathbf{Ku} = \mathbf{f}$. Apply that program to solve the previous Exercise.

Hint: here is a possible way of organizing the inner loop:

```
ii=0; For [i=1,i<=n,i++,  If [i==k, Continue[]]; ii++;
           c=K[[i,k]]/pivot; fc[[ii]]=f[[i]]-c*f[[k]]; jj=0;
           For [j=1,j<=n,j++, If [j==k,Continue[]]; jj++;
                Kc[[ii,jj]]=K[[i,j]]-c*K[[k,j]]
              ];
         ];
Return[{Kc,fc}]
```

EXERCISE 10.3 [D:15] Explain the similarities and differences between superelement analysis and global-local FEM analysis.

EXERCISE 10.4 [A:20] If the superelement stiffness $\mathbf{K}$ is symmetric, the static condensation process can be viewed as a special case of the master-slave transformation method discussed in Chapter 8. To prove this, take exterior freedoms $\mathbf{u}_b$ as masters and interior freedoms $\mathbf{u}_i$ as slaves. Assume the master-slave transformation relation

$$\mathbf{u} = \begin{bmatrix} \mathbf{u}_b \\ \mathbf{u}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib} \end{bmatrix} [\, \mathbf{u}_b \,] - \begin{bmatrix} \mathbf{0} \\ -\mathbf{K}_{ii}^{-1}\mathbf{f}_i \end{bmatrix} \stackrel{\rm def}{=} \mathbf{T}\mathbf{u}_b - \mathbf{g}. \tag{E10.3}$$

Work out $\hat{\mathbf{K}} = \mathbf{T}^T\mathbf{K}\mathbf{T}$ and $\hat{\mathbf{f}} = \mathbf{T}^T(\mathbf{f} - \mathbf{K}\mathbf{g})$, and show that $\hat{\mathbf{K}}\hat{\mathbf{u}} = \hat{\mathbf{f}}$ coalesces with the condensed stiffness equations (10.4)–(10.5) if $\mathbf{u}_b \equiv \hat{\mathbf{u}}$. (Take advantage of the symmetry properties $\mathbf{K}_{bi}^T = \mathbf{K}_{ib}$, $(\mathbf{K}_{ii}^{-1})^T = \mathbf{K}_{ii}^{-1}$.)

EXERCISE 10.5 [D:30] (Requires thinking) Explain the conceptual and operational differences between one-stage FEM analysis and global-local analysis of a problem such as that illustrated in Figures 10.6 and 10.7. Are the answers the same? What is gained, if any, by the global-local approach over the one-stage FEM analysis?

EXERCISE 10.6 [N:20] The widely used Guyan's scheme [326] for dynamic model reduction applies the static-condensation relation (E10.3) as master-slave transformation to both the stiffness and the mass matrix of the superelement. Use this procedure to eliminate the second and third DOF of the mass-stiffness system:

$$\mathbf{M} = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 1 & 4 & 1 & 0 \\ 0 & 1 & 4 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \quad \mathbf{K} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{bmatrix}. \tag{E10.4}$$

Compute and compare the vibration frequencies of the eigensystems $(\mathbf{M} - \omega_i^2 \mathbf{K})\mathbf{v}_i = \mathbf{0}$ and $(\hat{\mathbf{M}} - \hat{\omega}_i^2 \hat{\mathbf{K}})\hat{\mathbf{v}}_i = \mathbf{0}$ before and after reduction.

Hints. The four original squared frequencies are the eigenvalues of $\mathbf{M}^{-1}\mathbf{K}$, which may be obtained, for example, with the Matlab `eig` function. The largest is 2, lowest 0. To perform the Guyan reduction it is convenient to reorder **M** and **K** so that rows and columns 2 and 3 become 3 and 4, respectively:

$$\omega^2 \begin{bmatrix} 2 & 0 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 4 & 1 \\ 0 & 1 & 1 & 4 \end{bmatrix} \begin{bmatrix} v_1 \\ v_4 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_4 \\ v_2 \\ v_3 \end{bmatrix}, \tag{E10.5}$$

which in partitioned matrix form is

$$\omega^2 \begin{bmatrix} \mathbf{M}_{bb} & \mathbf{M}_{bi} \\ \mathbf{M}_{ib} & \mathbf{M}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{v}_b \\ \mathbf{v}_i \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{bb} & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_{ii} \end{bmatrix} \begin{bmatrix} \mathbf{v}_b \\ \mathbf{v}_i \end{bmatrix}. \tag{E10.6}$$

Next show that static condensation of **K** is equivalent to a master-slave transformation

$$\mathbf{T} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ib} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix} \quad \text{relating} \quad \begin{bmatrix} v_1 \\ v_4 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 2/3 & 1/3 \\ 1/3 & 2/3 \end{bmatrix} \begin{bmatrix} v_1 \\ v_4 \end{bmatrix}. \tag{E10.7}$$

The rest is easy. Form $\hat{\mathbf{M}} = \mathbf{T}^T\mathbf{M}\mathbf{T}$ and $\hat{\mathbf{K}} = \mathbf{T}^T\mathbf{K}\mathbf{T}$, find the two squared frequencies of the condensed eigensystem and verify that they approximate the two lowest original squared frequencies.

EXERCISE 10.7 [A:20] Two beam elements: 1–2 and 2–3, each of length L and rigidity EI are connected at node 2. The macroelement has 6 degrees of freedom: $\{v_1, \theta_1, v_2, \theta_2, v_3, \theta_3\}$. Eliminate the two DOF of node 2 by condensation. Is the condensed stiffness the same as that of a beam element of length $2L$ and rigidity EI?

(For expressions of the beam stiffness matrices, see Chapter 12.)