1
Overview

TABLE OF CONTENTS

This book is an introduction to the analysis of linear elastic structures by the Finite Element Method (FEM). This Chapter presents an overview of where the book fits, and what finite elements are.

§1.1. Where this Material Fits

The field of Mechanics can be subdivided into three major areas:

$$\text{Mechanics} \left\{ \begin{array}{l} \textit{Theoretical} \\ \textit{Applied} \\ \textit{Computational} \end{array} \right. \tag{1.1}$$

Theoretical mechanics deals with fundamental laws and principles of mechanics studied for their intrinsic scientific value. *Applied mechanics* transfers this theoretical knowledge to scientific and engineering applications, especially as regards the construction of mathematical models of physical phenomena. *Computational mechanics* solves specific problems by simulation through numerical methods implemented on digital computers.

Remark 1.1. Paraphrasing an old joke about mathematicians, one may define a computational mechanician as a person who searches for solutions to given problems, an applied mechanician as a person who searches for problems that fit given solutions, and a theoretical mechanician as a person who can prove the existence of problems and solutions.

§1.1.1. Computational Mechanics

Several branches of computational mechanics can be distinguished according to the *physical scale* of the focus of attention:

$$\text{Computational Mechanics} \left\{ \begin{array}{l} \textit{Nanomechanics and micromechanics} \\ \textit{Continuum mechanics} \left\{ \begin{array}{l} \text{Solids and Structures} \\ \text{Fluids} \\ \text{Multiphysics} \end{array} \right. \\ \textit{Systems} \end{array} \right. \tag{1.2}$$

Nanomechanics deals with phenomena at the molecular and atomic levels of matter. As such it is closely linked to particle physics and chemistry. Micromechanics looks primarily at the crystallographic and granular levels of matter. Its main technological application is the design and fabrication of materials and microdevices.

Continuum mechanics studies bodies at the macroscopic level, using continuum models in which the microstructure is homogenized by phenomenological averages. The two traditional areas of application are solid and fluid mechanics. The former includes *structures* which, for obvious reasons, are fabricated with solids. Computational solid mechanics takes an applied sciences approach, whereas computational structural mechanics emphasizes technological applications to the analysis and design of structures.

Computational fluid mechanics deals with problems that involve the equilibrium and motion of liquid and gases. Well developed subsidiaries are hydrodynamics, aerodynamics, acoustics, atmospheric physics, shock, combustion and propulsion.

Multiphysics is a more recent newcomer. This area is meant to include mechanical systems that transcend the classical boundaries of solid and fluid mechanics, as in interacting fluids and structures. Phase change problems such as ice melting and metal solidification fit into this category, as do the interaction of control, mechanical and electromagnetic systems.

Finally, *system* identifies mechanical objects, whether natural or artificial, that perform a distinguishable function. Examples of man-made systems are airplanes, buildings, bridges, engines, cars, microchips, radio telescopes, robots, roller skates and garden sprinklers. Biological systems, such as a whale, amoeba, inner ear, or pine tree are included if studied from the viewpoint of biomechanics. Ecological, astronomical and cosmological entities also form systems.[1]

In the progression of (1.2) the *system* is the most general concept. A system is studied by *decomposition*: its behavior is that of its components plus the interaction between components. Components are broken down into subcomponents and so on. As this hierarchical breakdown process continues, individual components become simple enough to be treated by individual disciplines, but component interactions get more complex. Consequently there is a tradeoff art in deciding where to stop.[2]

§1.1.2. Statics vs. Dynamics

Continuum mechanics problems may be subdivided according to whether inertial effects are taken into account or not:

$$\text{Continuum mechanics} \begin{cases} \textit{Statics} \\ \textit{Dynamics} \end{cases} \tag{1.3}$$

In dynamics actual time dependence must be explicitly considered, because the calculation of inertial (and/or damping) forces requires derivatives respect to actual time to be taken.

Problems in statics may also be time dependent but with inertial forces ignored or neglected. Accordingly static problems may be classed into strictly static and quasi-static. For the former time need not be considered explicitly; any historical time-like response-ordering parameter, if one is needed, will do. In quasi-static problems such as foundation settlement, metal creep, rate-dependent plasticity or fatigue cycling, a realistic measure of time is required but inertial forces are still neglected.

§1.1.3. Linear vs. Nonlinear

A classification of static problems that is particularly relevant to this book is

$$\text{Statics} \begin{cases} \textit{Linear} \\ \textit{Nonlinear} \end{cases} \tag{1.4}$$

Linear static analysis deals with static problems in which the *response* is linear in the cause-and-effect sense. For example: if the applied forces are doubled, the displacements and internal stresses also double. Problems outside this domain are classified as nonlinear.

---

[1] Except that their function may not be clear to us. "The usual approach of science of constructing a mathematical model cannot answer the questions of why there should be a universe for the model to describe. Why does the universe go to all the bother of existing? Is the unified theory so compelling that it brings about its own existence? Or does it need a creator, and, if so, does he have any other effect on the universe? And who created him?" (Stephen Hawking).

[2] Thus in breaking down a car engine for engineering analysis, say, the decomposition does not usually proceed beyond the components you can buy at a parts shop.

§1.1.4. Discretization methods

A final classification of CSM static analysis is based on the discretization method by which the continuum mathematical model is *discretized* in space, *i.e.*, converted to a discrete model with a finite number of degrees of freedom:

$$\text{Spatial discretization method} \left\{ \begin{array}{l} \textit{Finite Element (FEM)} \\ \textit{Boundary Element (BEM)} \\ \textit{Finite Difference (FDM)} \\ \textit{Finite Volume (FVM)} \\ \textit{Spectral} \\ \textit{Meshfree} \end{array} \right. \tag{1.5}$$

In CSM *linear* problems finite element methods currently dominate the scene as regards space discretization.[3] Boundary element methods post a strong second choice in specific application areas. For *nonlinear* problems the dominance of finite element methods is overwhelming.

Space finite difference methods in solid and structural mechanics have virtually disappeared from practical use. This statement is not true, however, for fluid mechanics, where finite difference discretization methods are still important. Finite-volume methods, which directly address the discretization of conservation laws, are important in difficult problems of fluid mechanics, for example high-Re gas dynamics. Spectral methods are based on transforms that map space and/or time dimensions to spaces (for example, the frequency domain) where the problem is easier to solve.

A recent newcomer to the scene are the meshfree methods. These combine techniques and tools of finite element methods such as variational formulation and interpolation, with finite difference features such as non-local support.

§1.1.5. FEM Variants

The term *Finite Element Method* actually identifies a broad spectrum of techniques that share common features outlined in §1.3 and §1.4. Two subclassifications that fit well applications to structural mechanics are[4]

$$\text{FEM Formulation} \left\{ \begin{array}{l} \textit{Displacement} \\ \textit{Equilibrium} \\ \textit{Mixed} \\ \textit{Hybrid} \end{array} \right. \qquad \text{FEM Solution} \left\{ \begin{array}{l} \textit{Stiffness} \\ \textit{Flexibility} \\ \textit{Mixed (a.k.a. Combined)} \end{array} \right. \tag{1.6}$$

Using the foregoing classification, we can state the topic of this book more precisely: the *computational analysis of linear static structural problems* by the Finite Element Method. Of the variants listed in (1.6), emphasis is placed on the *displacement* formulation and *stiffness* solution. This combination is called the *Direct Stiffness Method* or DSM.

---

[3] There are finite element discretizations in time, but they are not so widely used as finite differences.

[4] The distinction between these subclasses require advanced technical concepts, which cannot be covered in an introductory treatment such as this book.

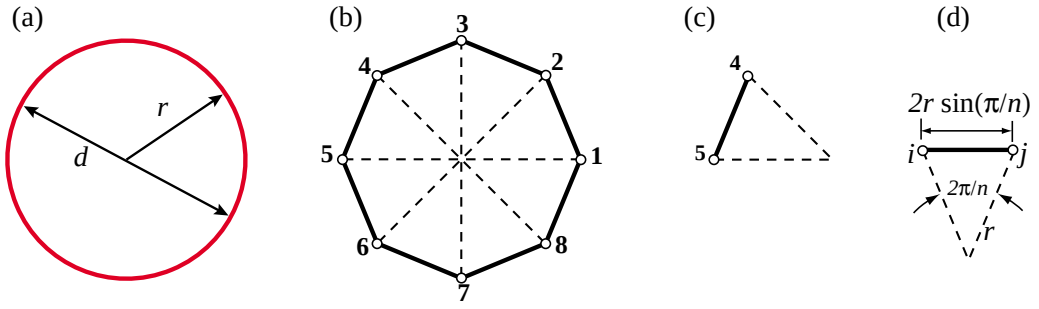

FIGURE 1.1. The "find π" problem treated with FEM concepts: (a) continuum object, (b) a discrete approximation by inscribed regular polygons, (c) disconnected element, (d) generic element.

§1.2. What Does a Finite Element Look Like?

The subject of this book is FEM. But what is a finite element? The concept will be partly illustrated through a truly ancient problem: find the perimeter L of a circle of diameter d. Since $L = \pi\, d$, this is equivalent to obtaining a numerical value for π.

Draw a circle of radius r and diameter $d = 2r$ as in Figure 1.1(a). Inscribe a regular polygon of n sides, where $n = 8$ in Figure 1.1(b). Rename polygon sides as *elements* and vertices as *nodes*. Label nodes with integers $1, \ldots 8$. Extract a typical element, say that joining nodes 4–5, as shown in Figure 1.1(c). This is an instance of the *generic element* i–j pictured in Figure 1.1(d). The element length is $L_{ij} = 2r \sin(\pi/n)$. Since all elements have the same length, the polygon perimeter is $L_n = nL_{ij}$, whence the approximation to π is $\pi_n = L_n/d = n \sin(\pi/n)$.

Table 1.1. Rectification of Circle by Inscribed Polygons ("Archimedes FEM")

n	$\pi_n = n \sin(\pi/n)$	Extrapolated by Wynn-ϵ	Exact π to 16 places
1	0.000000000000000		
2	2.000000000000000		
4	2.828427124746190	3.414213562373096	
8	3.061467458920718		
16	3.121445152258052	3.141418327933211	
32	3.136548490545939		
64	3.140331156954753	3.141592658918053	
128	3.141277250932773		
256	3.141513801144301	3.141592653589786	3.141592653589793

Values of π_n obtained for $n = 1, 2, 4, \ldots 256$ are listed in the second column of Table 1.1. As can be seen the convergence to π is fairly slow. However, the sequence can be transformed by Wynn's ϵ algorithm[5] into that shown in the third column. The last value displays 15-place accuracy.

Some key ideas behind the FEM can be identified in this example. The circle, viewed as a *source mathematical object*, is replaced by polygons. These are *discrete approximations* to the circle. The sides, renamed as *elements*, are specified by their end *nodes*. Elements can be separated by

[5] A widely used lozenge extrapolation algorithm that speeds up the convergence of many sequences. See, e.g, [190].

disconnecting nodes, a process called *disassembly* in the FEM. Upon disassembly a *generic element* can be defined, *independently of the original circle*, by the segment that connects two nodes i and j. The relevant element property: side length L_{ij}, can be computed in the generic element independently of the others, a property called *local support* in the FEM. The target property: the polygon perimeter, is obtained by reconnecting n elements and adding up their length; the corresponding steps in the FEM being *assembly* and *solution*, respectively. There is of course nothing magic about the circle; the same technique can be be used to rectify any smooth plane curve.[6]

This example has been offered in the FEM literature, e.g. in [117], to aduce that finite element ideas can be traced to Egyptian mathematicians from *circa* 1800 B.C., as well as Archimedes' famous studies on circle rectification by 250 B.C. But comparison with the modern FEM, as covered in following Chapters, shows this to be a stretch. The example does not illustrate the concept of degrees of freedom, conjugate quantities and local-global coordinates. It is guilty of circular reasoning: the compact formula $\pi = \lim_{n\to\infty} n \sin(\pi/n)$ uses the unknown π in the right hand side.[7] Reasonable people would argue that a circle is a simpler object than, say, a 128-sided polygon. Despite these flaws the example is useful in one respect: showing a fielder's choice in the replacement of one mathematical object by another. This is at the root of the simulation process described below.

§1.3. The FEM Analysis Process

Processes using FEM involve carrying out a sequence of steps in some way. Those sequences take two canonical configurations, depending on (i) the environment in which FEM is used and (ii) the main objective: model-based simulation of physical systems, or numerical approximation to mathematical problems. Both are reviewed below to introduce terminology used in the sequel.

§1.3.1. The Physical FEM

A canonical use of FEM is simulation of physical systems. This must done by using models, and so the process is often called *model-based simulation*.

The process is illustrated in Figure 1.2. The centerpiece is the *physical system* to be modeled. Accordingly, this configuration is called the *Physical FEM*. The processes of idealization and discretization are carried out *concurrently* to produce the discrete model. The solution step is handled by an equation solver often customized to FEM, which delivers a discrete solution (or solutions).

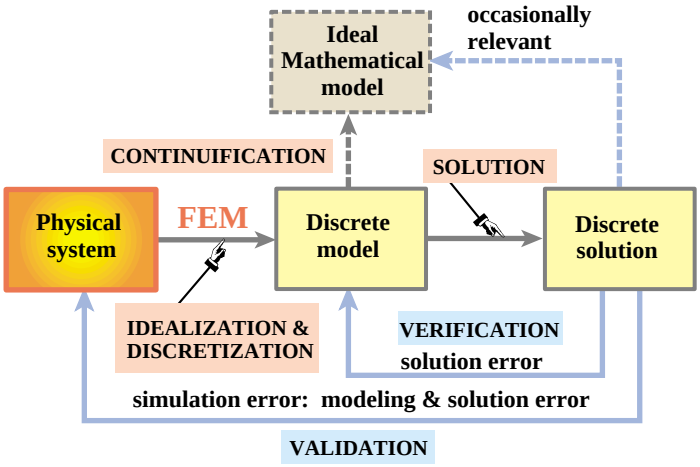

FIGURE 1.2. The Physical FEM. The physical system (left) is the source of the simulation process. The ideal mathematical model (should one go to the trouble of constructing it) is inessential.

[6] A similar limit process, however, may fail in three or more dimensions.

[7] This objection is bypassed if n is advanced as a power of two, as in Table 1.1, by using the half-angle recursion $\sqrt{2}\sin\alpha = \sqrt{1-\sqrt{1-\sin^2 2\alpha}}$, started from $2\alpha = \pi$ for which $\sin\pi = -1$.

Figure 1.2 also shows an *ideal mathematical model*. This may be presented as a *continuum limit* or "continuification" of the discrete model. For some physical systems, notably those well modeled by continuum fields, this step is useful. For others, such as complex engineering systems (say, a flying aircraft) it makes no sense. Indeed Physical FEM discretizations may be constructed and adjusted *without reference to mathematical models*, simply from experimental measurements.

The concept of *error* arises in the Physical FEM in two ways. These are known as *verification* and *validation*, respectively. Verification is done by replacing the discrete solution into the discrete model to get the solution error. This error is not generally important. Substitution in the ideal mathematical model in principle provides the discretization error. This step is rarely useful in complex engineering systems, however, because there is no reason to expect that the continuum model exists, and even if it does, that it is more physically relevant than the discrete model.

Validation tries to compare the discrete solution against observation by computing the *simulation error*, which combines modeling and solution errors. As the latter is typically unimportant, the simulation error in practice can be identified with the modeling error.

One way to adjust the discrete model so that it represents the physics better is called *model updating*. The discrete model is given free parameters. These are determined by comparing the discrete solution against experiments, as illustrated in Figure 1.3. Inasmuch as the minimization conditions are generally nonlinear (even if the model is linear) the updating process is inherently iterative.

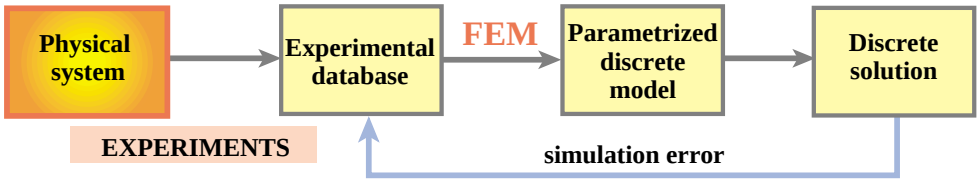

FIGURE 1.3. Model updating process in the Physical FEM.

§1.3.2. The Mathematical FEM

The other canonical way of using FEM focuses on the mathematics. The process steps are illustrated in Figure 1.4. The spotlight now falls on the *mathematical model*. This is often an ordinary or partial differential equation in space and time. A discrete finite element model is generated from a variational or weak form of the mathematical model.[8] This is the *discretization* step. The FEM equations are solved as described for the Physical FEM.

On the left Figure 1.4 shows an *ideal physical system*. This may be presented as a *realization* of the mathematical model. Conversely, the mathematical model is said to be an *idealization* of this system. E.g., if the mathematical model is the Poisson's PDE, realizations may be heat conduction or an electrostatic charge-distribution problem. This step is inessential and may be left out. Indeed Mathematical FEM discretizations *may be constructed without any reference to physics*.

The concept of *error* arises when the discrete solution is substituted in the "model" boxes. This replacement is generically called *verification*. As in the Physical FEM, the *solution error* is the

[8] The distinction between strong, weak and variational forms is discussed in advanced FEM courses. In the present book such forms will be largely stated (and used) as recipes.

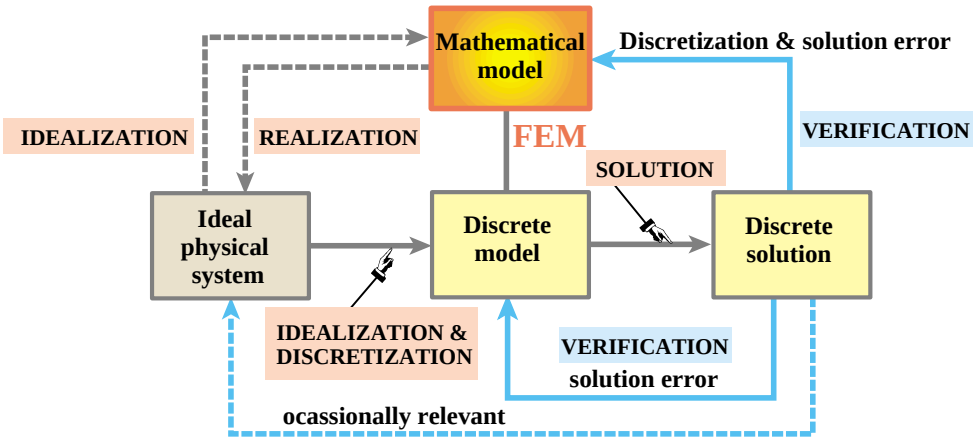

FIGURE 1.4. The Mathematical FEM. The mathematical model (top) is the source of the simulation process. Discrete model and solution follow from it. The ideal physical system (should one go to the trouble of exhibiting it) is inessential.

amount by which the discrete solution fails to satisfy the discrete equations. This error is relatively unimportant when using computers, and in particular direct linear equation solvers, for the solution step. More relevant is the *discretization error*, which is the amount by which the discrete solution fails to satisfy the mathematical model.[9] Replacing into the ideal physical system would in principle quantify modeling errors. In the Mathematical FEM this is largely irrelevant, however, because the ideal physical system is merely that: a figment of the imagination.

§1.3.3. Synergy of Physical and Mathematical FEM

The foregoing canonical sequences are not exclusive but complementary. This synergy[10] is one of the reasons behind the power and acceptance of the method. Historically the Physical FEM was the first one to be developed to model complex physical systems such as aircraft, as narrated in §1.7. The Mathematical FEM came later and, among other things, provided the necessary theoretical underpinnings to extend FEM beyond structural analysis.

A glance at the schematics of a commercial jet aircraft makes obvious the reasons behind the Physical FEM. There is no simple differential equation that captures, at a continuum mechanics level,[11] the structure, avionics, fuel, propulsion, cargo, and passengers eating dinner. There is no reason for despair, however. The time honored *divide and conquer* strategy, coupled with *abstraction*, comes to the rescue. First, separate the structure out and view the rest as masses and forces, most of which are time-varying and nondeterministic.

---

[9] This error can be computed in several ways, the details of which are of no importance here.

[10] Such interplay is not exactly a new idea: "The men of experiment are like the ant, they only collect and use; the reasoners resemble spiders, who make cobwebs out of their own substance. But the bee takes the middle course: it gathers its material from the flowers of the garden and field, but transforms and digests it by a power of its own." (Francis Bacon).

[11] Of course at the (sub)atomic level quantum mechanics works for everything, from landing gears to passengers. But it would be slightly impractical to represent the aircraft by, say, 10^{36} interacting particles modeled by the Schrödinger equations. More seriously, Truesdell and Toupin correctly note that "*Newtonian mechanics, while not appropriate to the corpuscles making up a body, agrees with experience when applied to the body as a whole*, except for certain phenomena of astronomical scale" [172, p. 228].

Second, consider the aircraft structure as built of *substructures* (a part of a structure devoted to a specific function): wings, fuselage, stabilizers, engines, landing gears, and so on. Take each substructure, and continue to break it down into *components*: rings, ribs, spars, cover plates, actuators, etc, continuing through as many levels as necessary.

Eventually those components become sufficiently simple in geometry and connectivity that they can be reasonably well described by the continuum mathematical models provided, for instance, by Mechanics of Materials or the Theory of Elasticity. At that point, *stop*. The component level discrete equations are obtained from a FEM library based on the mathematical model.

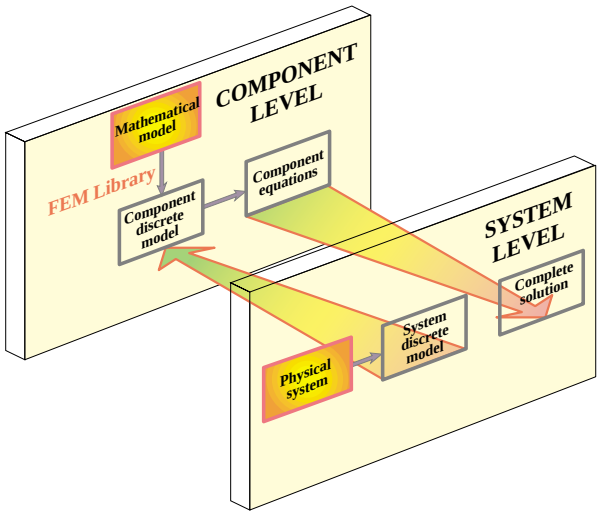

FIGURE 1.5. Combining physical and mathematical modeling through multilevel FEM. Only two levels (system and component) are shown for simplicity; intermediate substructure levels are omitted.

The system model is obtained by going through the reverse process: from component equations to substructure equations, and from those to the equations of the complete aircraft.

This *system assembly* process is governed by the classical principles of Newtonian mechanics, which provide the necessary "component glue." The multilevel decomposition process is diagramed in Figure 1.5, in which intermediate levels are omitted for simplicity.

Remark 1.2. More intermediate decomposition levels are used in systems such as offshore and ship structures, which are characterized by a modular fabrication process. In that case the multilevel decomposition mimics the way the system is actually fabricated. The general technique, called *superelements*, is discussed in Chapter 11.

Remark 1.3. There is no point in practice in going beyond a certain component level while considering the complete system. The reason is that the level of detail can become overwhelming without adding relevant information. Usually that point is reached when uncertainty impedes further progress. Further refinement of specific components is done by the so-called global-local analysis technique outlined in Chapter 11. This technique is an instance of *multiscale analysis*.

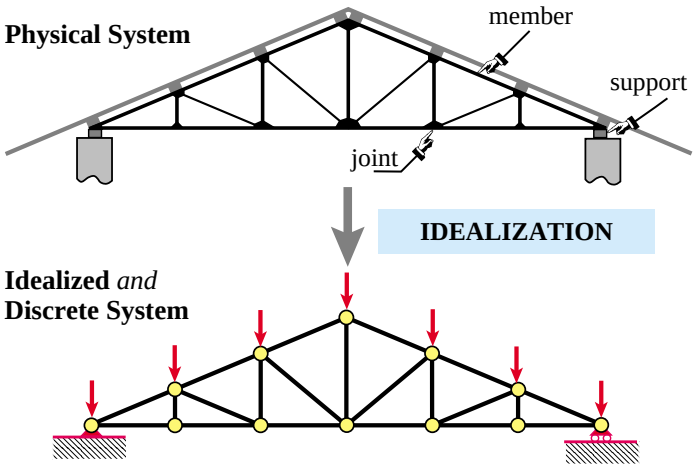

FIGURE 1.6. The idealization process for a simple structure. The physical system — here a roof truss — is directly idealized by the mathematical model: a pin-jointed bar assembly. For this particular structure idealized and discrete models coalesce.

For sufficiently simple structures, passing to a discrete model is carried out in a single *idealization and discretization* step, as illustrated for the truss roof structure shown in Figure 1.6. Other levels are unnecessary in such cases. Of course the truss may be viewed as a substructure of the roof, and the roof as a a substructure of a building.

§1.4. Interpretations of the Finite Element Method

Just like there are two complementary ways of using the FEM, there are two complementary interpretations for teaching it. One stresses the physical significance and is aligned with the Physical FEM. The other focuses on the mathematical context, and is aligned with the Mathematical FEM.

§1.4.1. Physical Interpretation

The physical interpretation focuses on the flowchart of Figure 1.2. This interpretation has been shaped by the discovery and extensive use of the method in the field of structural mechanics. The historical connection is reflected in the use of structural terms such as "stiffness matrix", "force vector" and "degrees of freedom," a terminology that carries over to non-structural applications.

The basic concept in the physical interpretation is the *breakdown* ($\equiv$ disassembly, tearing, partition, separation, decomposition) of a complex mechanical system into simpler, disjoint components called finite elements, or simply *elements*. The mechanical response of an element is characterized in terms of a finite number of degrees of freedom. These degrees of freedoms are represented as the values of the unknown functions as a set of node points. The element response is defined by algebraic equations constructed from mathematical or experimental arguments. The response of the original system is considered to be approximated by that of the *discrete model* constructed by *connecting* or *assembling* the collection of all elements.

The breakdown-assembly concept occurs naturally when an engineer considers many artificial and natural systems. For example, it is easy and natural to visualize an engine, bridge, aircraft or skeleton as being fabricated from simpler parts.

As discussed in §1.3, the underlying theme is *divide and conquer*. If the behavior of a system is too complex, the recipe is to divide it into more manageable subsystems. If these subsystems are still too complex the subdivision process is continued until the behavior of each subsystem is simple enough to fit a mathematical model that represents well the knowledge level the analyst is interested in. In the finite element method such "primitive pieces" are called *elements*. The behavior of the total system is that of the individual elements plus their interaction. A key factor in the initial acceptance of the FEM was that the element interaction can be physically interpreted and understood in terms that were eminently familiar to structural engineers.

§1.4.2. Mathematical Interpretation

This interpretation is closely aligned with the flowchart of Figure 1.4. The FEM is viewed as a procedure for obtaining numerical approximations to the solution of boundary value problems (BVPs) posed over a domain Ω. This domain is replaced by the union $\cup$ of disjoint subdomains $\Omega^{(e)}$ called finite elements. In general the geometry of Ω is only approximated by that of $\cup\Omega^{(e)}$.

The unknown function (or functions) is locally approximated over each element by an interpolation formula expressed in terms of values taken by the function(s), and possibly their derivatives, at a set of *node points* generally located on the element boundaries. The states of the assumed unknown function(s) determined by unit node values are called *shape functions*. The union of shape functions "patched" over adjacent elements form a *trial function basis* for which the node values represent the generalized coordinates. The trial function space may be inserted into the governing equations and the unknown node values determined by the Ritz method (if the solution extremizes a variational

principle) or by the Galerkin, least-squares or other weighted-residual minimization methods if the problem cannot be expressed in a standard variational form.

Remark 1.4. In the mathematical interpretation the emphasis is on the concept of *local (piecewise) approximation*. The concept of element-by-element breakdown and assembly, while convenient in the computer implementation, is not theoretically necessary. The mathematical interpretation permits a general approach to the questions of convergence, error bounds, trial and shape function requirements, etc., which the physical approach leaves unanswered. It also facilitates the application of FEM to classes of problems that are not so readily amenable to physical visualization as structures; for example electromagnetics and thermal conduction.

Remark 1.5. It is interesting to note some similarities in the development of Heaviside's operational methods, Dirac's delta-function calculus, and the FEM. These three methods appeared as ad-hoc computational devices created by engineers and physicists to deal with problems posed by new science and technology (electricity, quantum mechanics, and delta-wing aircraft, respectively) with little help from the mathematical establishment. Only some time after the success of the new techniques became apparent were new branches of mathematics (operational calculus, distribution theory and piecewise-approximation theory, respectively) constructed to justify that success. In the case of the finite element method, the development of a formal mathematical theory started in the late 1960s, and much of it is still in the making.

§1.5. Keeping the Course

The first Part of this book, covered in Chapters 2 through 11, stresses the physical interpretation of FEM within the framework of the Direct Stiffness Method (DSM). This is done on account of its instructional advantages. Furthermore the computer implementation becomes more transparent because the sequence of operations can be placed in close correspondence with the DSM steps.

Chapters 12 through 19 incorporate ingredients of the mathematical interpretation when it is felt convenient to do so. Nonetheless the exposition avoids excessive entanglement with the mathematical theory when it may obfuscate the physics.

In Chapters 2 and 3 the time is frozen at about 1965, and the DSM presented as an aerospace engineer of that time would have understood it. This is not done for sentimental reasons, although that happens to be the year in which the writer began thesis work on FEM under Ray Clough. Virtually all commercial codes are now based on the DSM and the computer implementation has not essentially changed since the late 1960s.[12] What has greatly improved since is "marketing sugar": user interaction and visualization.

§1.6. *What is Not Covered

The following topics are not covered in this book:

1. Elements based on equilibrium, mixed and hybrid variational formulations.
2. Flexibility and mixed solution methods of solution.
3. Kirchhoff-based plate and shell elements.
4. Continuum-based plate and shell elements.
5. Variational methods in mechanics.
6. General mathematical theory of finite elements.

[12] With the gradual disappearance of Fortran as a "live" programming language, noted in §1.7.7, changes at the computer implementation level have recently accelerated. For example C++ "wrappers" are becoming more common.

7. Buckling and stability analysis.
8. General nonlinear response analysis.
9. Structural optimization.
10. Error estimates and problem-adaptive discretizations.
11. Non-structural and multiphysics applications of FEM.
12. Designing and building production-level FEM software and use of special hardware (*e.g.* vector and parallel computers)

Topics 1–6 belong to what may be called "Advanced Linear FEM", whereas 7–8 pertain to "Nonlinear FEM". Topics 9–11 fall into advanced applications, whereas 12 is an interdisciplinary topic that interweaves with computer science.

§1.7. *Historical Sketch and Bibliography

This section summarizes the history of structural finite elements since 1950 to date. It functions as a hub for chapter-dispersed historical references.

For exposition convenience, structural "finitelementology" may be divided into four generations that span 10 to 15 years each. There are no sharp intergenerational breaks, but noticeable change of emphasis. The following summary does not cover the conjoint evolution of Matrix Structural Analysis into the Direct Stiffness Method from 1934 through 1970. This was the subject of a separate essay [59], which is also given in Appendix H.

§1.7.1. Who Invented Finite Elements?

Not just one individual, as this historical sketch will make clear. But if the question is tweaked to: who created the FEM in everyday use? there is no question in the writer's mind: M. J. (Jon) Turner at Boeing over the period 1950–1962. He formulated and perfected the Direct Stiffness Method, and forcefully got Boeing to commit resources to it while other aerospace companies were enmeshed in the Force Method. He established and formulated the first continuum based finite elements. In addition to Turner, major contributors to current practice include: B. M. Irons, inventor of isoparametric models, shape functions, the patch test and frontal solvers; R. J. Melosh, who recognized the Rayleigh-Ritz link and systematized the variational derivation of stiffness elements; and E. L. Wilson, who developed the first open source (and widely imitated) FEM software.

All of these pioneers were in the aerospace industry at least during part of their careers. That is not coincidence. FEM is the confluence of three ingredients, one of which is digital computation. And only large industrial companies (as well as some government agencies) were able to afford mainframe computers during the 1950s.

Who were the popularizers? Four academicians: J. H. Argyris, R. W. Clough, H. C. Martin, and O. C. Zienkiewicz are largely responsible for the "technology transfer" from the aerospace industry to a wider range of engineering applications during the 1950s and 1960s. The first three learned the method from Turner directly or indirectly. As a consultant to Boeing in the early 1950s, Argyris, a Force Method expert then at Imperial College, received reports from Turner's group, and weaved the material into his influencial 1954 serial [4]. Clough and Martin, then junior professors at U.C. Berkeley and U. Washington, respectively, spent "faculty internship" summers at Turner's group during 1952 and 1953. The result of this seminal collaboration was a celebrated paper [174], widely considered the start of the present FEM. Clough baptized the method in 1960 [26] and went on to form at Berkeley the first research group that propelled the idea into Civil Engineering applications. Olek Zienkiewicz, originally an expert in finite difference methods who learned the trade from Southwell, was convinced in 1964 by Clough to try FEM. He went on to write the first textbook on the subject [193] and to organize another important Civil Engineering research group in the University of Wales at Swansea.

§1.7.2. G1: The Pioneers

The 1956 paper by Turner, Clough, Martin and Topp [174], henceforth abbreviated to TCMT, is recognized as the start of the current FEM, as used in the overwhelming majority of commercial codes. Along with Argyris'

serial [4] they prototype the first generation, which spans 1950 through 1962. A panoramic picture of this period is available in two textbooks [130,140]. Przemieniecki's text is still reprinted by Dover. The survey by Gallagher [73] was influential at the time but is now difficult to access outside libraries.

The pioneers were structural engineers, schooled in classical mechanics. They followed a century of tradition in regarding structural elements as a device to transmit forces. This "element as force transducer" was the standard view in pre-computer structural analysis. It explains the use of flux assumptions to derive stiffness equations in TCMT. Element developers worked in, or interacted closely with, the aircraft industry. (As noted above, only large aerospace companies were then able to afford mainframe computers.) Accordingly they focused on thin structures built up with bars, ribs, spars, stiffeners and panels. Although the Classical Force Method dominated stress analysis during the 1950s [59], stiffness methods were kept alive by use in dynamics and vibration. It is not coincidence that Turner was an world-class expert in aeroelasticity.

§1.7.3. G2: The Golden Age

The next period spans the golden age of FEM: 1962–1972. This is the "variational generation." Melosh showed [121] that conforming displacement models are a form of Rayleigh-Ritz based on the minimum potential energy principle. This influential paper marks the confluence of three lines of research: Argyris' dual formulation of energy methods [4], the Direct Stiffness Method (DSM) of Turner [175–177], and early ideas of interelement compatibility as basis for error bounding and convergence [68,120]. G1 workers thought of finite elements as idealizations of structural components. From 1962 onward a two-step interpretation emerges: discrete elements approximate continuum models, which in turn approximate real structures.

By the early 1960s FEM begins to expand into Civil Engineering through Clough's Boeing-Berkeley connection [32,33] and had been baptized [26,28]. Reading Fraeijs de Veubeke's famous article [69] side by side with TCMT [174] one can sense the ongoing change in perspective opened up by the variational framework. The first book devoted to FEM appears in 1967 [193]. Applications to nonstructural problems had started in 1965 [192], and were treated in some depth by Martin and Carey [117].

From 1962 onwards the displacement formulation dominates. This was given a big boost by the invention of the isoparametric formulation and related tools (numerical integration, fitted natural coordinates, shape functions, patch test) by Irons and coworkers [100–103]. Low order displacement models often exhibit disappointing performance. Thus there was a frenzy to develop higher order elements. Other variational formulations, notably hybrids [133,136], mixed [93,163] and equilibrium models [69] emerged. G2 can be viewed as closed by the monograph of Strang and Fix [155], the first book to focus on the mathematical foundations.

§1.7.4. G3: Consolidation

The post-Vietnam economic doldrums are mirrored during this post-1972 period. Gone is the youthful exuberance of the golden age. This is consolidation time. Substantial effort is put into improving the stock of G2 displacement elements by tools initially labeled "variational crimes" [154], but later justified. Textbooks by Hughes [99] and Bathe [9] reflect the technology of this period. Hybrid and mixed formulations record steady progress [8]. Assumed strain formulations appear [110]. A booming activity in error estimation and mesh adaptivity is fostered by better understanding of the mathematical foundations [161].

Commercial FEM codes gradually gain importance. They provide a reality check on what works in the real world and what doesn't. By the mid-1980s there was gathering evidence that complex and high order elements were commercial flops. Exotic gadgetry interweaved amidst millions of lines of code easily breaks down in new releases. Complexity is particularly dangerous in nonlinear and dynamic analyses conducted by novice users. A trend back toward simplicity starts [111,112].

§1.7.5. G4: Back to Basics

The fourth generation begins by the early 1980s. More approaches come on the scene, notably the Free Formulation [16,17], orthogonal hourglass control [64], Assumed Natural Strain methods [10,151], stress hybrid

models in natural coordinates [131,141], as well as variants and derivatives of those approaches: ANDES [50,122], EAS [147,148] and others. Although technically diverse the G4 approaches share two common objectives:

(i) Elements must fit into DSM-based programs since that includes the vast majority of production codes, commercial or otherwise.

(ii) Elements are kept simple but should provide answers of engineering accuracy with relatively coarse meshes. These were collectively labeled "high performance elements" in 1989 [49].

"Things are always at their best in the beginning," said Pascal. Indeed. By now FEM looks like an aggregate of largely disconnected methods and recipes. The blame should not be placed on the method itself, but on the community split noted in the book Preface.

2

The Direct Stiffness Method I

TABLE OF CONTENTS

This Chapter begins the exposition of the Direct Stiffness Method (DSM) of structural analysis. The DSM is by far the most common implementation of the Finite Element Method (FEM). In particular, all major commercial FEM codes are based on the DSM.

The exposition is done by following the DSM steps applied to a simple plane truss structure. The method has two major stages: breakdown, and assembly+solution. This Chapter covers primarily the breakdown stage.

§2.1. Why A Plane Truss?

The simplest structural finite element is the 2-node bar (also called linear spring) element, which is illustrated in Figure 2.1(a). Perhaps the most complicated finite element (at least as regards number of degrees of freedom) is the curved, three-dimensional "brick" element depicted in Figure 2.1(b).

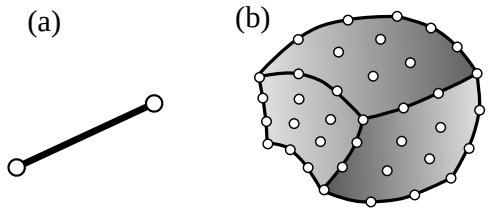

FIGURE 2.1. From the simplest through a highly complex structural finite element: (a) 2-node bar element for trusses, (b) 64-node tricubic, "brick" element for three-dimensional solid analysis.

Yet the remarkable fact is that, in the DSM, the simplest and most complex elements are treated alike! To illustrate the basic steps of this democratic method, it makes educational sense to keep it simple and use a structure composed of bar elements.

A simple yet nontrivial structure is the *pin-jointed plane truss*.[1] Using a plane truss to teach the stiffness method offers two additional advantages:

(a) Computations can be entirely done by hand as long as the structure contains just a few elements. This allows various steps of the solution procedure to be carefully examined and understood before passing to the computer implementation. Doing hand computations on more complex finite element systems rapidly becomes impossible.

(b) The computer implementation on any programming language is relatively simple and can be assigned as preparatory computer homework before reaching Part III.

§2.2. Truss Structures

Plane trusses, such as the one depicted in Figure 2.2, are often used in construction, particularly for roofing of residential and commercial buildings, and in short-span bridges. Trusses, whether two or three dimensional, belong to the class of *skeletal structures*. These structures consist of elongated structural components called *members*, connected at *joints*. Another important subclass of skeletal structures are frame structures or *frameworks*, which are common in reinforced concrete construction of buildings and bridges.

Skeletal structures can be analyzed by a variety of hand-oriented methods of structural analysis taught in beginning Mechanics of Materials courses: the Displacement and Force methods. They can also be analyzed by the computer-oriented FEM. That versatility makes those structures a good choice

[1] A one dimensional bar assembly would be even simpler. That kind of structure would not adequately illustrate some of the DSM steps, however, notably the back-and-forth transformations from global to local coordinates.

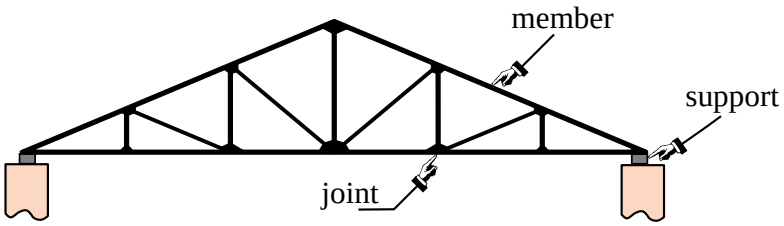

FIGURE 2.2. An actual plane truss structure. That shown is typical of a roof truss used in building construction.

to illustrate the transition from the hand-calculation methods taught in undergraduate courses, to the fully automated finite element analysis procedures available in commercial programs.

In this and the next Chapter we will go over the basic steps of the DSM in a "hand-computer" calculation mode. This means that although the steps are done by hand, whenever there is a procedural choice we shall either adopt the way which is better suited towards the computer implementation, or explain the difference between hand and computer computations. The actual computer implementation using a high-level programming language is presented in Chapter 5.

To keep hand computations manageable in detail we use just about the simplest structure that can be called a plane truss, namely the three-member truss illustrated in Figure 2.3. The *idealized* model of the example truss as a pin-jointed assemblage of bars is shown in Figure 2.4(a), which also gives its geometric and material properties. In this idealization truss members carry only axial loads, have no bending resistance, and are connected by frictionless pins. Figure 2.4(b) displays support conditions as well as the applied forces applied to the truss joints.

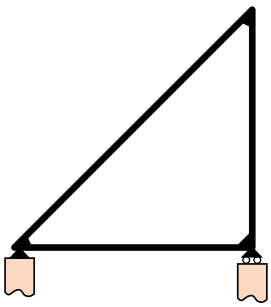

FIGURE 2.3. The three-member example truss.

It should be noted that as a practical structure the example truss is not particularly useful — the one depicted in Figure 2.2 is far more common in construction. But with the example truss we can go over the basic DSM steps without getting mired into too many members, joints and degrees of freedom.

§2.3. Idealization

Although the pin-jointed assemblage of bars (as depicted in Figure 2.4) is sometimes presented as an actual problem, it actually represents an *idealization* of a true truss structure. The axially-carrying members and frictionless pins of this structure are only an approximation of a real truss. For example, building and bridge trusses usually have members joined to each other through the use of gusset plates, which are attached by nails, bolts, rivets or welds. See Figure 2.2. Consequently members will carry some bending as well as direct axial loading.

Experience has shown, however, that stresses and deformations calculated for the simple idealized problem will often be satisfactory for overall-design purposes; for example to select the cross section of the members. Hence the engineer turns to the pin-jointed assemblage of axial force elements and uses it to carry out the structural analysis.

This replacement of true by idealized is at the core of the *physical interpretation* of the finite element method discussed in §1.4.

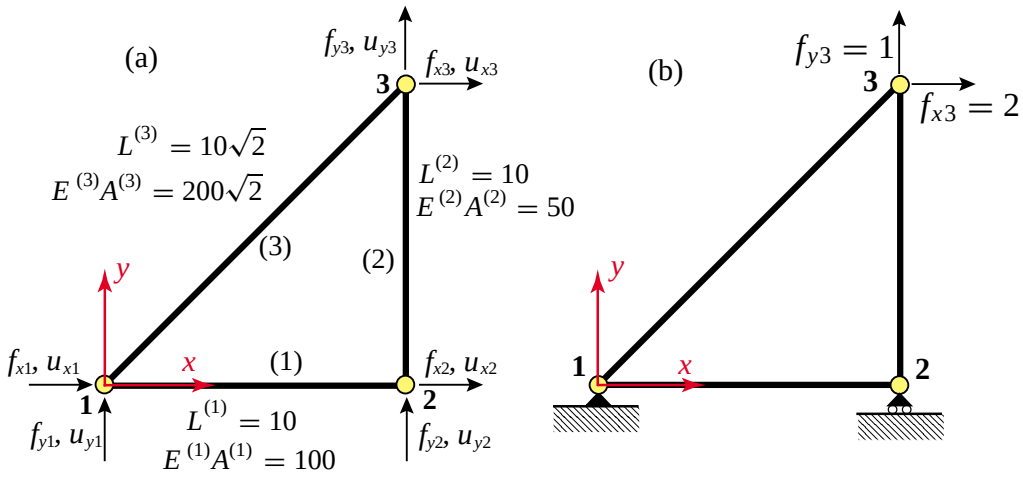

FIGURE 2.4. Pin-jointed idealization of example truss: (a) geometric and elastic properties, (b) support conditions and applied loads.

§2.4. Members, Joints, Forces and Displacements

The idealization of the example truss, pictured in Figure 2.4, has three *joints*, which are labeled 1, 2 and 3, and three *members*, which are labeled (1), (2) and (3). These members connect joints 1–2, 2–3, and 1–3, respectively. The member lengths are denoted by $L^{(1)}$, $L^{(2)}$ and $L^{(3)}$, their elastic moduli by $E^{(1)}$, $E^{(2)}$ and $E^{(3)}$, and their cross-sectional areas by $A^{(1)}$, $A^{(2)}$ and $A^{(3)}$. Note that an element number supercript is enclosed in parenthesis to avoid confusion with exponents. Both E and A are assumed to be constant along each member.

Members are generically identified by index e (because of their close relation to finite elements, see below). This index is placed as supercript of member properties. For example, the cross-section area of a generic member is A^e. The member superscript is *not* enclosed in parentheses in this case because no confusion with exponents can arise. But the area of member 3 is written $A^{(3)}$ and not A^3.

Joints are generically identified by indices such as i, j or n. In the general FEM, the name "joint" and "member" is replaced by *node* and *element*, respectively. The dual nomenclature is used in the initial Chapters to stress the physical interpretation of the FEM.

The geometry of the structure is referred to a common Cartesian coordinate system $\{x, y\}$, which is called the *global coordinate system*. Other names for it in the literature are *structure coordinate system* and *overall coordinate system*.

The key ingredients of the stiffness method of analysis are the *forces* and *displacements* at the joints.

In a idealized pin-jointed truss, externally applied forces as well as reactions *can act only at the joints*. All member axial forces can be characterized by the x and y components of these forces, denoted by f_x and f_y, respectively. The components at joint i will be identified as f_{xi} and f_{yi}, respectively. The set of all joint forces can be arranged as a 6-component column vector called **f**.

The other key ingredient is the displacement field. Classical structural mechanics tells us that the displacements of the truss *are completely defined by the displacements of the joints*. This statement is a particular case of the more general finite element theory. The x and y displacement components will be denoted by u_x and u_y, respectively. The *values* of u_x and u_y at joint i will be called u_{xi} and u_{yi}. Like joint forces, they are arranged into a 6-component vector called **u**. Here are the two vectors

of nodal forces and nodal displacements, shown side by side:

$$\mathbf{f} = \begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{bmatrix}, \qquad \mathbf{u} = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix}. \tag{2.1}$$

In the DSM these six displacements are the primary unknowns. They are also called the *degrees of freedom* or *state variables* of the system.[2]

How about the displacement boundary conditions, popularly called support conditions? This data will tell us which components of **f** and **u** are actual unknowns and which ones are known *a priori*. In pre-computer structural analysis such information was used *immediately* by the analyst to discard unnecessary variables and thus reduce the amount of hand-carried bookkeeping.

The computer oriented philosophy is radically different: *boundary conditions can wait until the last moment*. This may seem strange, but on the computer the sheer volume of data may not be so important as the efficiency with which the data is organized, accessed and processed. The strategy "save the boundary conditions for last" will be followed here also for the hand computations.

Remark 2.1. Often column vectors such as (2.1) will be displayed in row form to save space, with a transpose symbol at the end. For example, $\mathbf{f} = [\ f_{x1}\ \ f_{y1}\ \ f_{x2}\ \ f_{y2}\ \ f_{x3}\ \ f_{y3}\ ]^T$ and $\mathbf{u} = [\ u_{x1}\ \ u_{y1}\ \ u_{x2}\ \ u_{y2}\ \ u_{x3}\ \ u_{y3}\ ]^T$.

§2.5. The Master Stiffness Equations

The *master stiffness equations* relate the joint forces **f** of the complete structure to the joint displacements **u** of the complete structure *before* specification of support conditions.

Because the assumed behavior of the truss is linear, these equations must be linear relations that connect the components of the two vectors. Furthermore it will be assumed that if all displacements vanish, so do the forces.[3] If both assumptions hold the relation must be homogeneous and expressable in component form as

$$\begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{bmatrix} = \begin{bmatrix} K_{x1x1} & K_{x1y1} & K_{x1x2} & K_{x1y2} & K_{x1x3} & K_{x1y3} \\ K_{y1x1} & K_{y1y1} & K_{y1x2} & K_{y1y2} & K_{y1x3} & K_{y1y3} \\ K_{x2x1} & K_{x2y1} & K_{x2x2} & K_{x2y2} & K_{x2x3} & K_{x2y3} \\ K_{y2x1} & K_{y2y1} & K_{y2x2} & K_{y2y2} & K_{y2x3} & K_{y2y3} \\ K_{x3x1} & K_{x3y1} & K_{x3x2} & K_{x3y2} & K_{x3x3} & K_{x3y3} \\ K_{y3x1} & K_{y3y1} & K_{y3x2} & K_{y3y2} & K_{y3x3} & K_{y3y3} \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix}. \tag{2.2}$$

In matrix notation:

$$\boxed{\mathbf{f} = \mathbf{K}\,\mathbf{u}.} \tag{2.3}$$

---

[2] *Primary unknowns* is the correct mathematical term whereas *degrees of freedom* has a mechanics flavor: "any of a limited number of ways in which a body may move or in which a dynamic system may change" (Merrian-Webster). The term *state variables* is used more often in nonlinear analysis, material sciences and statistics.

[3] This assumption implies that the so-called *initial strain* effects, also known as *prestress* or *initial stress* effects, are neglected. Such effects are produced by actions such as temperature changes or lack-of-fit fabrication, and are studied in Chapter 29.

Here **K** is the *master stiffness matrix*, also called *global stiffness matrix*, *assembled stiffness matrix*, or *overall stiffness matrix*. It is a 6×6 square matrix that happens to be symmetric, although this attribute has not been emphasized in the written-out form (2.2). The entries of the stiffness matrix are often called *stiffness coefficients* and have a physical interpretation discussed below.

The qualifiers ("master", "global", "assembled" and "overall") convey the impression that there is another level of stiffness equations lurking underneath. And indeed there is a *member level* or *element level*, into which we plunge in the **Breakdown** section.

Remark 2.2. *Interpretation of Stiffness Coefficients*. The following interpretation of the entries of **K** is valuable for visualization and checking. Choose a displacement vector **u** such that all components are zero except the i^{th} one, which is one. Then **f** is simply the i^{th} column of **K**. For instance if in (2.3) we choose u_{x2} as unit displacement,

$$\mathbf{u} = [\,0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0\,]^T, \qquad \mathbf{f} = [\,K_{x1x2} \;\; K_{y1x2} \;\; K_{x2x2} \;\; K_{y2x2} \;\; K_{x3x2} \;\; K_{y3x2}\,]^T. \tag{2.4}$$

Thus K_{y1x2}, say, represents the y-force at joint 1 that would arise on prescribing a unit x-displacement at joint 2, while all other displacements vanish. In structural mechanics this property is called *interpretation of stiffness coefficients* as *displacement influence coefficients*. It extends unchanged to the general finite element method.

§2.6. The DSM Steps

The DSM steps, major and minor, are summarized in Figure 2.5 for the convenience of the reader. The two major processing steps are **Breakdown**, followed by **Assembly & Solution**. A postprocessing substep may follow, although this is not part of the DSM proper.

The first 3 DSM substeps are: (1) disconnection, (2) localization, and (3) computation of member stiffness equations. Collectively these form the *breakdown*. The first two are marked as *conceptual* in Figure 2.5 because they are not actually programmed as such. These subsets are implicitly carried out through the user-provided problem definition. Processing begins at the member-stiffness-equation forming substep.

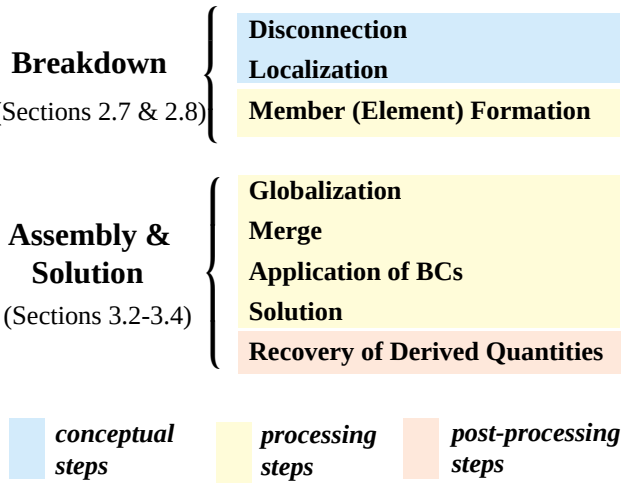

FIGURE 2.5. The Direct Stiffness Method steps.

§2.7. Breakdown

§2.7.1. Disconnection

To carry out the first breakdown step we proceed to *disconnect* or *disassemble* the structure into its components, namely the three truss members. This task is illustrated in Figure 2.6. To each member $e = 1, 2, 3$ assign a Cartesian system $\{\bar{x}^e, \bar{y}^e\}$. Axis $\bar{x}^e$ is aligned along the axis of the e^{th} member. Actually $\bar{x}^e$ runs along the member longitudinal axis; it is shown offset in that Figure for clarity.

By convention the positive direction of $\bar{x}^e$ runs from joint i to joint j, where $i < j$. The angle formed by $\bar{x}^e$ and x is the *orientation angle* φ^e. The axes origin is arbitrary and may be placed at the member midpoint or at one of the end joints for convenience.

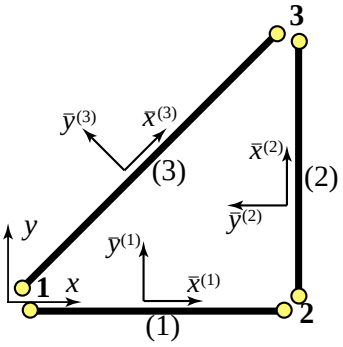

FIGURE 2.6. Breakdown of example truss into individual members (1), (2) and (3), and selection of local coordinate systems.

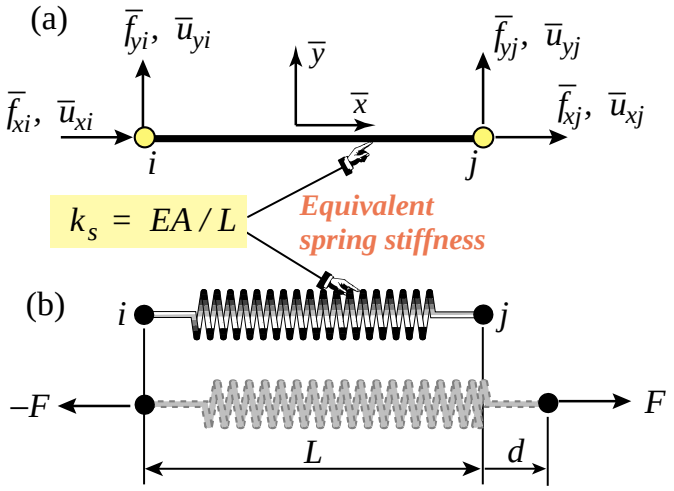

FIGURE 2.7. Generic truss member referred to its local coordinate system $\{\bar{x}, \bar{y}\}$: (a) idealization as bar element, (b) interpretation as equivalent spring.

Systems $\{\bar{x}^e, \bar{y}^e\}$ are called *local coordinate systems* or *member-attached coordinate systems*. In the general finite element method they also receive the name *element coordinate systems*.

§2.7.2. Localization

Next we drop the member identifier e so that we are effectively dealing with a *generic* truss member, as illustrated in Figure 2.7(a). The local coordinate system is $\{\bar{x}, \bar{y}\}$. The two end joints are i and j.

As shown in that figure, a generic truss member has four joint force components and four joint displacement components (the member degrees of freedom). The member properties are length L, elastic modulus E and cross-section area A.

§2.7.3. Computation of Member Stiffness Equations

The force and displacement components of the generic truss member shown in Figure 2.7(a) are linked by the *member stiffness relations*

$$\bar{\mathbf{f}} = \bar{\mathbf{K}}\,\bar{\mathbf{u}}, \tag{2.5}$$

which written out in full is

$$\begin{bmatrix} \bar{f}_{xi} \\ \bar{f}_{yi} \\ \bar{f}_{xj} \\ \bar{f}_{yj} \end{bmatrix} = \begin{bmatrix} \bar{K}_{xixi} & \bar{K}_{xiyi} & \bar{K}_{xixj} & \bar{K}_{xiyj} \\ \bar{K}_{yixi} & \bar{K}_{yiyi} & \bar{K}_{yixj} & \bar{K}_{yiyj} \\ \bar{K}_{xjxi} & \bar{K}_{xjyi} & \bar{K}_{xjxj} & \bar{K}_{xjyj} \\ \bar{K}_{yjxi} & \bar{K}_{yjyi} & \bar{K}_{yjxj} & \bar{K}_{yjyj} \end{bmatrix} \begin{bmatrix} \bar{u}_{xi} \\ \bar{u}_{yi} \\ \bar{u}_{xj} \\ \bar{u}_{yj} \end{bmatrix}. \tag{2.6}$$

Vectors $\bar{\mathbf{f}}$ and $\bar{\mathbf{u}}$ are called the *member joint forces* and *member joint displacements*, respectively, whereas $\bar{\mathbf{K}}$ is the *member stiffness matrix* or *local stiffness matrix*. When these relations are interpreted from the standpoint of the general FEM, "member" is replaced by "element" and "joint" by "node."

There are several ways to construct the stiffness matrix $\bar{\mathbf{K}}$ in terms of L, E and A. The most straightforward technique relies on the Mechanics of Materials approach covered in undergraduate

courses. Think of the truss member in Figure 2.7(a) as a linear spring of equivalent stiffness k_s, an interpretation illustrated in Figure 2.7(b). If the member properties are *uniform* along its length, Mechanics of Materials bar theory tells us that[4]

$$k_s = \frac{EA}{L}, \tag{2.7}$$

Consequently the force-displacement equation is

$$F = k_s d = \frac{EA}{L} d, \tag{2.8}$$

where F is the internal axial force and d the relative axial displacement, which physically is the bar elongation. The axial force and elongation can be immediately expressed in terms of the joint forces and displacements as

$$F = \bar{f}_{xj} = -\bar{f}_{xi}, \qquad d = \bar{u}_{xj} - \bar{u}_{xi}, \tag{2.9}$$

which express force equilibrium[5] and kinematic compatibility, respectively. Combining (2.8) and (2.9) we obtain the matrix relation[6]

$$\bar{\mathbf{f}} = \begin{bmatrix} \bar{f}_{xi} \\ \bar{f}_{yi} \\ \bar{f}_{xj} \\ \bar{f}_{yj} \end{bmatrix} = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{u}_{xi} \\ \bar{u}_{yi} \\ \bar{u}_{xj} \\ \bar{u}_{yj} \end{bmatrix} = \bar{\mathbf{K}}\,\bar{\mathbf{u}}, \tag{2.10}$$

Hence

$$\boxed{\bar{\mathbf{K}} = \frac{EA}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.} \tag{2.11}$$

This is the truss stiffness matrix in local coordinates.

Two other methods for obtaining the local force-displacement relation (2.8) are covered in Exercises 2.6 and 2.7.

§2.8. Assembly: Globalization

The first substep in the assembly & solution major step, as shown in Figure 2.5, is *globalization*. This operation is done member by member. It refers the member stiffness equations to the global system $\{x, y\}$ so it can be merged into the master stiffness. Before entering into details we must establish relations that connect joint displacements and forces in the global and local coordinate systems. These are given in terms of *transformation matrices*.

---

[4] See for example, Chapter 2 of [12].

[5] Equations $F = \bar{f}_{xj} = -\bar{f}_{xi}$ follow by considering the free body diagram (FBD) of each joint. For example, take joint i as a FBD. Equilibrium along x requires $-F - \bar{f}_{xi} = 0$ whence $F = -\bar{f}_{xi}$. Doing the same on joint j yields $F = \bar{f}_{xj}$.

[6] The matrix derivation of (2.10) is the subject of Exercise 2.3.

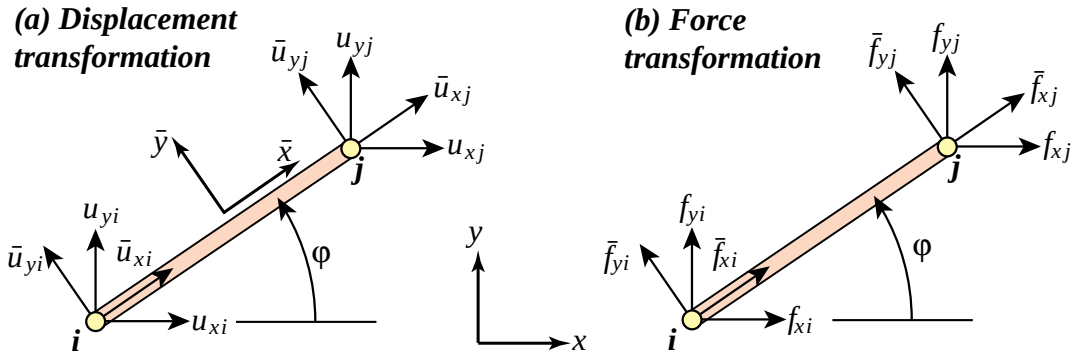

FIGURE 2.8. The transformation of node displacement and force components from the local system $\{\bar{x}, \bar{y}\}$ to the global system $\{x, y\}$.

§2.8.1. Coordinate Transformations

The necessary transformations are easily obtained by inspection of Figure 2.8. For the displacements

$$\begin{aligned} \bar{u}_{xi} &= u_{xi}c + u_{yi}s, \qquad \bar{u}_{yi} = -u_{xi}s + u_{yi}c, \\ \bar{u}_{xj} &= u_{xj}c + u_{yj}s, \qquad \bar{u}_{yj} = -u_{xj}s + u_{yj}c, \end{aligned} \tag{2.12}$$

where $c = \cos\varphi, s = \sin\varphi$ and φ is the angle formed by $\bar{x}$ and x, measured positive counterclockwise from x. The matrix form of this relation is

$$\begin{bmatrix} \bar{u}_{xi} \\ \bar{u}_{yi} \\ \bar{u}_{xj} \\ \bar{u}_{yj} \end{bmatrix} = \begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix} \begin{bmatrix} u_{xi} \\ u_{yi} \\ u_{xj} \\ u_{yj} \end{bmatrix}. \tag{2.13}$$

The 4×4 matrix that appears above is called a *displacement transformation matrix* and is denoted[7] by $\mathbf{T}$. The node forces transform as $f_{xi} = \bar{f}_{xi}c - \bar{f}_{yi}s$, etc., which in matrix form become

$$\begin{bmatrix} f_{xi} \\ f_{yi} \\ f_{xj} \\ f_{yj} \end{bmatrix} = \begin{bmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & c & -s \\ 0 & 0 & s & c \end{bmatrix} \begin{bmatrix} \bar{f}_{xi} \\ \bar{f}_{yi} \\ \bar{f}_{xj} \\ \bar{f}_{yj} \end{bmatrix}. \tag{2.14}$$

The 4×4 matrix that appears above is called a *force transformation matrix*. A comparison of (2.13) and (2.14) reveals that the force transformation matrix is the *transpose* $\mathbf{T}^T$ of the displacement transformation matrix $\mathbf{T}$. This relation is not accidental and can be proved to hold generally.[8]

Remark 2.3. Note that in (2.13) the local system (barred) quantities appear on the left-hand side, whereas in (2.14) they show up on the right-hand side. The expressions (2.13) and and (2.14) are discrete counterparts of what are called covariant and contravariant transformations, respectively, in continuum mechanics. The counterpart of the transposition relation is the *adjointness* property.

---

[7] This matrix will be called $\mathbf{T}_d$ when its association with displacements is to be emphasized, as in Exercise 2.5.

[8] A simple proof that relies on the invariance of external work is given in Exercise 2.5. However this invariance was only checked by explicit computation for a truss member in Exercise 2.4. The general proof relies on the Principle of Virtual Work, which is discussed later.

Remark 2.4. For this particular structural element $\mathbf{T}$ is square and orthogonal, that is, $\mathbf{T}^T = \mathbf{T}^{-1}$. But this property does not extend to more general elements. Furthermore in the general case $\mathbf{T}$ is not even a square matrix, and does not possess an ordinary inverse. However the congruential transformation relations (2.15)–(2.17) do hold generally.

§2.8.2. Transformation to Global System

From now on we reintroduce the member (element) index, e. The member stiffness equations in global coordinates will be written

$$\mathbf{f}^e = \mathbf{K}^e \mathbf{u}^e. \tag{2.15}$$

The compact form of (2.13) and (2.14) for the e^{th} member is

$$\bar{\mathbf{u}}^e = \mathbf{T}^e \mathbf{u}^e, \qquad \mathbf{f}^e = (\mathbf{T}^e)^T \bar{\mathbf{f}}^e. \tag{2.16}$$

Inserting these matrix expressions into $\bar{\mathbf{f}}^e = \overline{\mathbf{K}}^e \bar{\mathbf{u}}^e$ and comparing with (2.15) we find that the member stiffness in the global system $\{x, y\}$ can be computed from the member stiffness $\bar{\mathbf{K}}^e$ in the local system $\{\bar{x}, \bar{y}\}$ through the congruential transformation

$$\boxed{\mathbf{K}^e = (\mathbf{T}^e)^T \bar{\mathbf{K}}^e \mathbf{T}^e.} \tag{2.17}$$

Carrying out the matrix multiplications in closed form we get

$$\mathbf{K}^e = \frac{E^e A^e}{L^e} \begin{bmatrix} c^2 & sc & -c^2 & -sc \\ sc & s^2 & -sc & -s^2 \\ -c^2 & -sc & c^2 & sc \\ -sc & -s^2 & sc & s^2 \end{bmatrix}, \tag{2.18}$$

in which $c = \cos\varphi^e$, $s = \sin\varphi^e$, with e superscripts of c and s suppressed to reduce clutter. If the angle is zero we recover (2.10), as may be expected. $\mathbf{K}^e$ is called a *member stiffness matrix in global coordinates*. The proof of (2.17) and verification of (2.18) is left as Exercise 2.8.

The globalized member stiffness matrices for the example truss can now be easily obtained by inserting appropriate values into (2.18). For member (1), with end joints 1–2, angle $\varphi = 0^\circ$ and the member properties given in Figure 2.4(a) we get

$$\begin{bmatrix} f^{(1)}_{x1} \\ f^{(1)}_{y1} \\ f^{(1)}_{x2} \\ f^{(1)}_{y2} \end{bmatrix} = 10 \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^{(1)}_{x1} \\ u^{(1)}_{y1} \\ u^{(1)}_{x2} \\ u^{(1)}_{y2} \end{bmatrix}. \tag{2.19}$$

For member (2), with end joints 2–3, and angle $\varphi = 90^\circ$:

$$\begin{bmatrix} f^{(2)}_{x2} \\ f^{(2)}_{y2} \\ f^{(2)}_{x3} \\ f^{(2)}_{y3} \end{bmatrix} = 5 \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} u^{(2)}_{x2} \\ u^{(2)}_{y2} \\ u^{(2)}_{x3} \\ u^{(2)}_{y3} \end{bmatrix}. \tag{2.20}$$

Finally, for member (3), with end joints 1–3, and angle $\varphi = 45^\circ$:

$$\begin{bmatrix} f_{x1}^{(3)} \\ f_{y1}^{(3)} \\ f_{x3}^{(3)} \\ f_{y3}^{(3)} \end{bmatrix} = 20 \begin{bmatrix} 0.5 & 0.5 & -0.5 & -0.5 \\ 0.5 & 0.5 & -0.5 & -0.5 \\ -0.5 & -0.5 & 0.5 & 0.5 \\ -0.5 & -0.5 & 0.5 & 0.5 \end{bmatrix} \begin{bmatrix} u_{x1}^{(3)} \\ u_{y1}^{(3)} \\ u_{x3}^{(3)} \\ u_{y3}^{(3)} \end{bmatrix}. \tag{2.21}$$

In the following Chapter we will complete the main DSM steps by putting the truss back together through the merge step, and solving for the unknown forces and displacements.

Notes and Bibliography

The Direct Stiffness Method has been the dominant FEM version since the mid-1960s, and is the procedure followed by all major commercial codes in current use. DSM was invented and developed at Boeing in the early 1950s, through the leadership of Jon Turner [174–177], and had defeated its main competitor, the Force Method, by 1970 [59].

All applications-oriented FEM books cover the DSM, although the procedural steps are sometimes not clearly identified. In particular, the textbooks recommended in §1.7.6 offer adequate expositions.

Trusses, also called bar assemblies, are usually the first structures treated in Mechanics of Materials books written for undergraduate courses. Two widely used books at this level are [12] and [137].

Steps in the derivation of stiffness matrices for truss elements are well covered in a number of early treatment of finite element books, of which Chapter 5 of Przemieniecki [140] is a good example.

References

Referenced items have been moved to Appendix R.

3

The Direct Stiffness Method II

TABLE OF CONTENTS

§3.1. The Remaining DSM Steps

Chapter 2 covered the initial stages of the DSM. The three breakdown steps: *disconnection*, *localization* and *formation of member stiffness* take us down all the way to the generic truss element: the highest level of fragmentation. This is followed by the *assembly* process.

Assembly involves *merging* the stiffness equations of each member into the global stiffness equations. For this to make sense, the member equations must be referred to a common coordinate system, which for a plane truss is the global Cartesian system $\{x, y\}$. This is done through the globalization process covered in §2.8. On the computer the formation, globalization and merge steps are done concurrently, member by member. After all members are processed we have the *free-free master stiffness equations*.

Next comes the *solution*. This process embodies two substeps: *application of boundary conditions* and *solution* for the unknown joint displacements. To apply the BCs, the free-free master stiffness equations are modified by taking into account which components of the joint displacements and forces are given and which are unknown.

The modified equations are submitted to a linear equation solver, which returns the unknown joint (node) displacements. As discussed under **Notes and Bibliography**, on some FEM implementations — especially programs written in the 1960s and 1970s — one or more of the foregoing operations are done concurrently.

The solution step completes the DSM proper. *Postprocessing* steps may follow, in which derived quantities such as internal forces and stresses are recovered from the displacement solution.

§3.2. Assembly: Merge

§3.2.1. Governing Rules

The key operation of the assembly process is the "placement" of the contribution of each member to the master stiffness equations. The process is technically called *merge* of individual members. The merge operation can be physically interpreted as *reconnecting* that member in the process of fabricating the complete structure. For a truss structure, reconnection means inserting the pins back into the joints. See Figure 3.1.

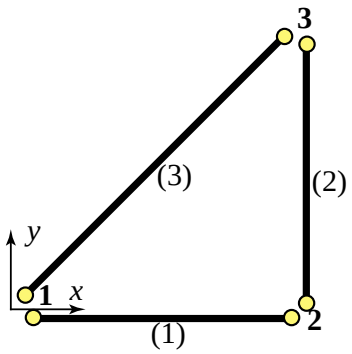

FIGURE 3.1. The disconnected example truss prior to merge. All member stiffness equations are in the global system. Reconnecting the truss means putting the pins back into the joints.

Merging logic is mathematically governed by two rules of structural mechanics:

1. *Compatibility of displacements*: The displacement of all members meeting at a joint are the same.
2. *Force equilibrium*: The sum of forces exerted by all members that meet at a joint balances the external force applied to that joint. (3.1)

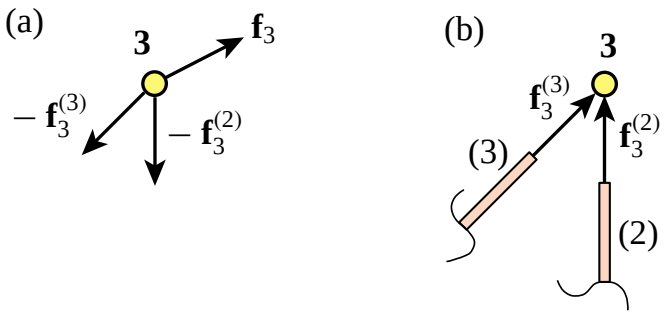

FIGURE 3.2. The force equilibrium of joint 3 of the example truss, depicted as a free body diagram in (a). Here $\mathbf{f}_3$ is the known external joint force applied on the joint. Joint forces $\mathbf{f}_3^{(2)}$ and $\mathbf{f}_3^{(3)}$ are applied *by the joint on the members*, as illustrated in (b). Consequently the forces applied *by the members on the joint* are $-\mathbf{f}_3^{(2)}$ and $-\mathbf{f}_3^{(3)}$. These forces would act in the directions shown in (a) if both members (2) and (3) were in tension. The free-body equilibrium statement is $\mathbf{f}_3 - \mathbf{f}_3^{(2)} - \mathbf{f}_3^{(3)} = \mathbf{0}$ or $\mathbf{f}_3 = \mathbf{f}_3^{(2)} + \mathbf{f}_3^{(3)}$. This translates into the two component equations: $f_{x3} = f_{x3}^{(2)} + f_{x3}^{(3)}$ and $f_{y3} = f_{y3}^{(2)} + f_{y3}^{(3)}$, of (3.2).

The first rule is physically obvious: reconnected joints must move as one entity. The second one can be visualized by considering a joint as a free body, although care is required in the interpretation of joint forces and their signs. Notational conventions to this effect are explained in Figure 3.2 for joint 3 of the example truss, at which members (2) and (3) meet. Application of the foregoing rules at this particular joint gives

$$\begin{aligned} &\text{Rule 1:} \qquad u_{x3}^{(2)} = u_{x3}^{(3)}, \quad u_{y3}^{(2)} = u_{y3}^{(3)}. \\ &\text{Rule 2:} \quad f_{x3} = f_{x3}^{(2)} + f_{x3}^{(3)} = f_{x3}^{(1)} + f_{x3}^{(2)} + f_{x3}^{(3)}, \quad f_{y3} = f_{y3}^{(2)} + f_{y3}^{(3)} = f_{y3}^{(1)} + f_{y3}^{(2)} + f_{y3}^{(3)}. \end{aligned} \tag{3.2}$$

The addition of $f_{x3}^{(1)}$ to $f_{x3}^{(2)} + f_{x3}^{(3)}$ and of and $f_{y3}^{(1)}$ to $f_{y3}^{(2)} + f_{y3}^{(3)}$, respectively, changes nothing because member (1) is not connected to joint 3. We are just adding zeros. But this augmentation enables us to write the key matrix relation:

$$\boxed{\mathbf{f} = \mathbf{f}^{(1)} + \mathbf{f}^{(2)} + \mathbf{f}^{(3)}.} \tag{3.3}$$

§3.2.2. Hand Assembly by Augmentation and Merge

To directly visualize how the two rules (3.1) translate to merging logic, we first *augment* the member stiffness relations by adding zero rows and columns as appropriate to *complete* the force and displacement vectors.

For member (1):

$$\begin{bmatrix} f_{x1}^{(1)} \\ f_{y1}^{(1)} \\ f_{x2}^{(1)} \\ f_{y2}^{(1)} \\ f_{x3}^{(1)} \\ f_{y3}^{(1)} \end{bmatrix} = \begin{bmatrix} 10 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{x1}^{(1)} \\ u_{y1}^{(1)} \\ u_{x2}^{(1)} \\ u_{y2}^{(1)} \\ u_{x3}^{(1)} \\ u_{y3}^{(1)} \end{bmatrix}. \tag{3.4}$$

For member (2):

$$\begin{bmatrix} f^{(2)}_{x1} \\ f^{(2)}_{y1} \\ f^{(2)}_{x2} \\ f^{(2)}_{y2} \\ f^{(2)}_{x3} \\ f^{(2)}_{y3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & -5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5 & 0 & 5 \end{bmatrix} \begin{bmatrix} u^{(2)}_{x1} \\ u^{(2)}_{y1} \\ u^{(2)}_{x2} \\ u^{(2)}_{y2} \\ u^{(2)}_{x3} \\ u^{(2)}_{y3} \end{bmatrix}. \tag{3.5}$$

For member (3):

$$\begin{bmatrix} f^{(3)}_{x1} \\ f^{(3)}_{y1} \\ f^{(3)}_{x2} \\ f^{(3)}_{y2} \\ f^{(3)}_{x3} \\ f^{(3)}_{y3} \end{bmatrix} = \begin{bmatrix} 10 & 10 & 0 & 0 & -10 & -10 \\ 10 & 10 & 0 & 0 & -10 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & 0 & 10 & 10 \end{bmatrix} \begin{bmatrix} u^{(3)}_{x1} \\ u^{(3)}_{y1} \\ u^{(3)}_{x2} \\ u^{(3)}_{y2} \\ u^{(3)}_{x3} \\ u^{(3)}_{y3} \end{bmatrix}. \tag{3.6}$$

According to the first rule, we can *drop the member identifier* in the displacement vectors that appear in the foregoing matrix equations. Hence the reconnected member equations are

$$\begin{bmatrix} f^{(1)}_{x1} \\ f^{(1)}_{y1} \\ f^{(1)}_{x2} \\ f^{(1)}_{y2} \\ f^{(1)}_{x3} \\ f^{(1)}_{y3} \end{bmatrix} = \begin{bmatrix} 10 & 0 & -10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix}, \tag{3.7}$$

$$\begin{bmatrix} f^{(2)}_{x1} \\ f^{(2)}_{y1} \\ f^{(2)}_{x2} \\ f^{(2)}_{y2} \\ f^{(2)}_{x3} \\ f^{(2)}_{y3} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & -5 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -5 & 0 & 5 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix}, \tag{3.8}$$

$$\begin{bmatrix} f^{(3)}_{x1} \\ f^{(3)}_{y1} \\ f^{(3)}_{x2} \\ f^{(3)}_{y2} \\ f^{(3)}_{x3} \\ f^{(3)}_{y3} \end{bmatrix} = \begin{bmatrix} 10 & 10 & 0 & 0 & -10 & -10 \\ 10 & 10 & 0 & 0 & -10 & -10 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & 0 & 10 & 10 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix}. \tag{3.9}$$

These three equations can be represented in direct matrix notation as

$$\mathbf{f}^{(1)} = \mathbf{K}^{(1)}\,\mathbf{u}, \qquad \mathbf{f}^{(2)} = \mathbf{K}^{(2)}\,\mathbf{u}, \qquad \mathbf{f}^{(3)} = \mathbf{K}^{(3)}\,\mathbf{u}. \tag{3.10}$$

According to the second rule, expressed in matrix form as (3.3), we have

$$\mathbf{f} = \mathbf{f}^{(1)} + \mathbf{f}^{(2)} + \mathbf{f}^{(3)} = \left(\mathbf{K}^{(1)} + \mathbf{K}^{(2)} + \mathbf{K}^{(3)}\right)\mathbf{u} = \mathbf{K}\,\mathbf{u}, \tag{3.11}$$

so all we have to do is add the three stiffness matrices that appear above, and we arrive at the master stiffness equations:

$$\boxed{\begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{bmatrix} = \begin{bmatrix} 20 & 10 & -10 & 0 & -10 & -10 \\ 10 & 10 & 0 & 0 & -10 & -10 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & -5 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & -5 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix}.} \tag{3.12}$$

Using this technique *member merging* becomes simply *matrix addition*.

This explanation of the assembly process is conceptually the easiest to follow and understand. It is virtually foolproof for hand computations. However, this is *not* the way the process is carried out on the computer because it would be enormously wasteful of storage for large systems. A computer-oriented procedure is discussed in §3.5.

§3.3. Solution

Having formed the master stiffness equations we can proceed to the solution phase. To prepare the equations for a linear solver we need to separate known and unknown components of **f** and **u**. In this Section a technique suitable for hand computation is described.

§3.3.1. Applying Displacement BCs by Reduction

If one attempts to solve the system (3.12) numerically for the displacements, surprise! The solution "blows up" because the coefficient matrix (the master stiffness matrix) is singular. The mathematical interpretation of this behavior is that rows and columns of **K** are linear combinations of each other (see Remark 3.1 below). The physical interpretation of singularity is that there are unsuppressed *rigid body motions*: the truss still "floats" in the $\{x, y\}$ plane.

To eliminate rigid body motions and render the system nonsingular we must apply the physical *support conditions* as *displacement boundary conditions*. From Figure 2.4(b) we observe that the support conditions for the example truss are

$$u_{x1} = u_{y1} = u_{y2} = 0, \tag{3.13}$$

whereas the known applied forces are

$$f_{x2} = 0, \quad f_{x3} = 2, \quad f_{y3} = 1. \tag{3.14}$$

When solving the overall stiffness equations by hand, the simplest way to account for support conditions is to *remove* equations associated with known joint displacements from the master system. To apply (3.13) we have to remove equations 1, 2 and 4. This can be systematically

accomplished by *deleting* or "striking out" rows and columns number 1, 2 and 4 from **K** and the corresponding components from **f** and **u**. The reduced three-equation system is

$$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 10 \\ 0 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} f_{x2} \\ f_{x3} \\ f_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix}. \tag{3.15}$$

Equation (3.15) is called the *reduced master stiffness system*. The coefficient matrix of this system is no longer singular.

Remark 3.1. In mathematical terms, the free-free master stiffness matrix **K** in (3.12) has order $N = 6$, rank $r = 3$ and a rank deficiency of $d = N - r = 6 - 3 = 3$ (these concepts are summarized in Appendix C.) The dimension of the null space of **K** is $d = 3$. This space is spanned by three independent rigid body motions: the two rigid translations along x and y and the rigid rotation about z.

Remark 3.2. Conditions (3.13) represent the simplest type of support conditions, namely specified zero displacements. More general constraint forms, such as prescribed nonzero displacements and multifreedom constraints, are handled as described in §3.6 and Chapters 8–9, respectively.

§3.3.2. Solving for Displacements

Solving the reduced system by hand (for example, via Gauss elimination) yields

$$\begin{bmatrix} u_{x2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0.4 \\ -0.2 \end{bmatrix}. \tag{3.16}$$

This is called a *partial displacement solution* (also *reduced displacement solution*) because it excludes known displacement components. This solution vector is *expanded* to six components by including the three specified values (3.13) in the appropiate slots:

$$\mathbf{u} = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.4 \\ -0.2 \end{bmatrix}. \tag{3.17}$$

This is the *complete displacement solution*, or simply the *displacement solution*.

§3.4. PostProcessing

The last processing step of the DSM is the solution for joint displacements. But often the analyst needs information on other mechanical quantities; for example the reaction forces at the supports, or the internal member forces. Such quantities are said to be *derived* because they are *recovered* from the displacement solution. The recovery of derived quantities is part of the so-called *postprocessing steps* of the DSM. Two such steps are described below.

§3.4.1. Recovery of Reaction Forces

Premultiplying the complete displacement solution (3.17) by **K** we get

$$\mathbf{f} = \mathbf{K}\mathbf{u} = \begin{bmatrix} 20 & 10 & -10 & 0 & -10 & -10 \\ 10 & 10 & 0 & 0 & -10 & -10 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & -5 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & -5 & 10 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0.4 \\ -0.2 \end{bmatrix} = \begin{bmatrix} -2 \\ -2 \\ 0 \\ 1 \\ 2 \\ 1 \end{bmatrix} \tag{3.18}$$

This vector recovers the known applied forces (3.14) as can be expected. Furthermore we get three *reaction forces*: $f_{x1} = f_{y1} = -2$ and $f_{y2} = 1$, which are associated with the support conditions (3.13). It is easy to check that the complete force system is in self equilibrium for the free-free structure; this is the topic of Exercise 3.1.

§3.4.2. Recovery of Internal Forces and Stresses

Often the structural engineer is not so much interested in displacements as in *internal forces* and *stresses*. These are in fact the most important quantities for preliminary structural design. In pin-jointed trusses the only internal forces are the *axial member forces*. For the example truss these forces, denoted by $F^{(1)}$, $F^{(2)}$ and $F^{(3)}$, are depicted in Figure 3.3. The average axial stress σ^e is obtained on dividing F^e by the cross-sectional area of the member.

The axial force F^e in member e can be obtained as follows. Extract the displacements of member e from the complete displacement solution **u** to form $\mathbf{u}^e$. Then recover local joint displacements from $\bar{\mathbf{u}}^e = \mathbf{T}^e\mathbf{u}^e$.

Compute the member elongation d^e (relative displacement) and recover the axial force from the equivalent spring constitutive relation:

$$d^e = \bar{u}^e_{xj} - \bar{u}^e_{xi}, \qquad F^e = \frac{E^e A^e}{L^e} d^e. \tag{3.19}$$

Note that $\bar{u}^e_{yi}$ and $\bar{u}^e_{yj}$ are not needed in computing d^e.

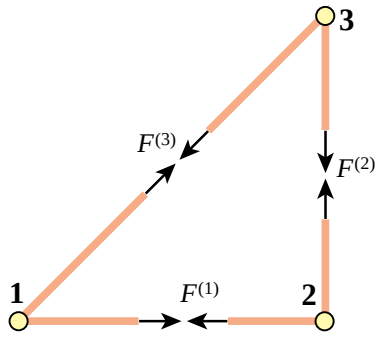

FIGURE 3.3. The internal forces in the example truss are the axial forces $F^{(1)}$, $F^{(2)}$ and $F^{(3)}$ in the members. Directions shown pertain to tension.

Example 3.1. Recover $F^{(2)}$ in example truss. Member (2) goes from node 2 to node 3 and $\varphi^{(2)} = 90^\circ$. Extract the global displacements of the member from (3.17): $\mathbf{u}^{(2)} = [\, u_{x2} \;\; u_{y2} \;\; u_{x3} \;\; u_{y3} \,]^T = [\, 0 \;\; 0 \;\; 0.4 \;\; 0.2 \,]^T$. Convert to local displacements using $\bar{\mathbf{u}}^{(2)} = \mathbf{T}^{(2)}\mathbf{u}^{(2)}$:

$$\begin{bmatrix} \bar{u}_{x2} \\ \bar{u}_{y2} \\ \bar{u}_{x3} \\ \bar{u}_{y3} \end{bmatrix} = \begin{bmatrix} \cos 90^\circ & \sin 90^\circ & 0 & 0 \\ -\sin 90^\circ & \cos 90^\circ & 0 & 0 \\ 0 & 0 & \cos 90^\circ & \sin 90^\circ \\ 0 & 0 & -\sin 90^\circ & \cos 90^\circ \end{bmatrix} \begin{bmatrix} u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0.4 \\ -0.2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -0.2 \\ -0.4 \end{bmatrix}. \tag{3.20}$$

Then $d^{(2)} = \bar{u}_{x3} - \bar{u}_{x2} = -0.2 - 0 = -0.2$, and $F^{(2)} = (50/10) \times -0.2 = -1$ (compression).

Remark 3.3. An alternative interpretation of (3.19) is to regard $e^e = d^e/L^e$ as the (average) member axial strain, $\sigma^e = E^e e^e$ as (average) axial stress, and $F^e = A^e\sigma^e$ as the axial force. This is more in tune with the Theory of Elasticity viewpoint discussed in Exercise 2.6.

§3.5. *Computer Oriented Assembly and Solution

§3.5.1. *Assembly by Freedom Pointers

The practical computer implementation of the DSM assembly process departs significantly from the "augment and add" technique described in §3.1.4. There are two major differences:

(I) Member stiffness matrices are *not* expanded. Their entries are directly merged into those of **K** through the use of a "freedom pointer array" called the *Element Freedom Table* or EFT.

(II) The master stiffness matrix **K** is stored using a special format that takes advantage of symmetry and sparseness.

Difference (II) is a more advanced topic that is deferred to the last part of the book. For simplicity we shall assume here that **K** is stored as a *full square matrix*, and study only (I). For the example truss the freedom-pointer technique expresses the entries of **K** as the sum

$$K_{pq} = \sum_{e=1}^{3} K_{ij}^{e} \qquad \text{for} \quad i = 1,\ldots 4,\;\; j = 1,\ldots 4,\;\; p = \text{EFT}^e(i),\;\; q = \text{EFT}^e(j). \tag{3.21}$$

Here K_{ij}^{e} denote the entries of the 4×4 globalized member stiffness matrices in (2.19) through (2.21). Entries K_{pq} that do not get any contributions from the right hand side remain zero. EFT^e denotes the Element Freedom Table for member e. For the example truss these tables are

$$\text{EFT}^{(1)} = \{1, 2, 3, 4\}, \qquad \text{EFT}^{(2)} = \{3, 4, 5, 6\}, \qquad \text{EFT}^{(3)} = \{1, 2, 5, 6\}. \tag{3.22}$$

Physically these tables map local freedom indices to global ones. For example, freedom number 3 of member (2) is u_{x3}, which is number 5 in the master equations; consequently $\text{EFT}^{(2)}(3) = 5$. Note that (3.21) involves three nested loops: over e (outermost), over i, and over j. The ordering of the last two is irrelevant. Advantage may be taken of the symmetry of $\mathbf{K}^e$ and **K** to roughly halve the number of additions. Exercise 3.5 follows the scheme (3.21) by hand.

The assembly process for general structures using this technique is studied in Chapter 25.

§3.5.2. *Applying DBC by Modification

In §3.3.1 the support conditions (3.13) were applied by reducing (3.12) to (3.15). Reduction is convenient for hand computations because it cuts down on the number of equations to solve. But it has a serious flaw for computer implementation: the equations must be rearranged. It was previously noted that on the computer the number of equations is not the only important consideration. Rearrangement can be as or more expensive than solving the equations, particularly if the coefficient matrix is stored in sparse form or on secondary storage.[1]

To apply support conditions without rearranging the equations we clear (set to zero) rows and columns corresponding to prescribed zero displacements as well as the corresponding force components, and place ones on the diagonal to maintain non-singularity. The resulting system is called the *modified* set of master stiffness equations. For the example truss this approach yields

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 10 \\ 0 & 0 & 0 & 0 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{bmatrix}, \tag{3.23}$$

[1] On most modern computers, reading a floating-point number from memory at a random address takes 100 to 1000 times as long as performing a floating-point arithmetic operation on numbers that are already in registers.

in which rows and columns for equations 1, 2 and 4 have been cleared. Solving this modified system produces the complete displacement solution (3.17) directly.

Remark 3.4. In a "smart" stiffness equation solver the modified system need not be explicitly constructed by storing zeros and ones. It is sufficient to *mark* the equations that correspond to displacement BCs. The solver is then programmed to skip those equations. However, if one is using a standard solver from, say, a library of scientific routines or a commercial program such as *Matlab* or *Mathematica*, such intelligence cannot be expected, and the modified system must be set up explicitly.

§3.6. Prescribed Nonzero Displacements

The support conditions considered in the example truss resulted in the specification of zero displacement components; for example $u_{y2} = 0$. There are cases, however, where the known value is nonzero. This happens, for example, in the study of settlement of foundations of ground structures such as buildings and bridges, and in the analysis of motion-driven machinery components.

Mathematically these are called non-homogenous boundary conditions. The treatment of this generalization of the FEM equations is studied in the following subsections.

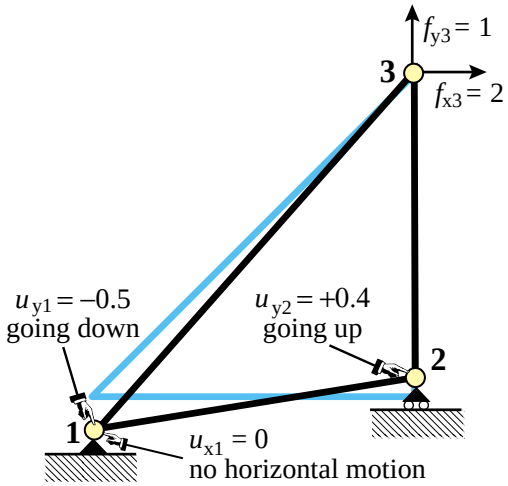

FIGURE 3.4. The example truss with prescribed nonzero vertical displacements at joints 1 and 2.

§3.6.1. Application of Nonzero-DBCs by Reduction

We describe first a matrix reduction technique, analogous to that used in §3.3.1, which is suitable for hand computations. Recall the master stiffness equations (3.12) for the example truss:

$$\begin{bmatrix} 20 & 10 & -10 & 0 & -10 & -10 \\ 10 & 10 & 0 & 0 & -10 & -10 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & -5 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & -5 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \\ f_{x3} \\ f_{y3} \end{bmatrix} \tag{3.24}$$

Suppose that the applied forces are again (3.14) but the prescribed displacements change to

$$u_{x1} = 0, \quad u_{y1} = -0.5, \quad u_{y2} = 0.4 \tag{3.25}$$

This means that joint 1 goes down vertically whereas joint 2 goes up vertically, as depicted in Figure 3.4. Inserting the known data into (3.24) we get

$$\begin{bmatrix} 20 & 10 & -10 & 0 & -10 & -10 \\ 10 & 10 & 0 & 0 & -10 & -10 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5 & 0 & -5 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & -5 & 10 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ -0.5 \\ u_{x2} \\ 0.4 \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} f_{x1} \\ f_{y1} \\ 0 \\ f_{y2} \\ 2 \\ 1 \end{bmatrix} \tag{3.26}$$

The first, second and fourth rows of (3.26) are removed, leaving only

$$\begin{bmatrix} -10 & 0 & 10 & 0 & 0 & 0 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & -5 & 10 & 15 \end{bmatrix} \begin{bmatrix} 0 \\ -0.5 \\ u_{x2} \\ 0.4 \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} \tag{3.27}$$

Columns 1, 2 and 4 are removed by transferring all known terms from the left to the right hand side:

$$\begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 10 \\ 0 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} - \begin{bmatrix} (-10)\times 0 + 0\times(-0.5) + 0\times 0.4 \\ (-10)\times 0 + (-10)\times(-0.5) + 0\times 0.4 \\ (-10)\times 0 + (-10)\times(-0.5) + (-5)\times 0.4 \end{bmatrix} = \begin{bmatrix} 0 \\ -3 \\ -2 \end{bmatrix}. \tag{3.28}$$

These are the *reduced stiffness equations*. Note that its coefficient matrix of (3.28) is exactly the same as in the reduced system (3.15) for prescribed zero displacements. The right hand side, however, is different. It consists of the applied joint forces *modified by the effect of known nonzero displacements*. These are called the *modified node forces* or *effective node forces*. Solving the reduced system yields

$$\begin{bmatrix} u_{x2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.5 \\ 0.2 \end{bmatrix}. \tag{3.29}$$

Filling the missing entries with the known values (3.25) yields the complete displacement solution (listed as row vector to save space):

$$\mathbf{u} = [\, 0 \quad -0.5 \quad 0 \quad 0.4 \quad -0.5 \quad 0.2 \,]^T . \tag{3.30}$$

Taking the solution (3.30) and going through the postprocessing steps discussed in §3.4, we can find that *reaction forces and internal member forces do not change*. This is a consequence of the fact that the example truss is *statically determinate*. The force systems (internal and external) in such structures are insensitive to movements such as foundation settlements.

§3.6.2. *Application of Nonzero-DBCs by Modification

The computer-oriented modifification approach follows the same idea outlined in §3.5.2. As there, the main objective is to avoid rearranging the master stiffness equations. To understand the process it is useful to think of being done in two stages. First equations 1, 2 and 4 are modified so that they become trivial equations, as illustrated for the example truss and the displacement boundary conditions (3.25):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -10 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ -10 & -10 & 0 & 0 & 10 & 10 \\ -10 & -10 & 0 & -5 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{x2} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0.4 \\ 2 \\ 1 \end{bmatrix} \tag{3.31}$$

The solution of this system recovers (3.26) by construction (for example, the fourth equation is simply $1\times u_{y2} = 0.4$). In the next stage, columns 1, 2 and 4 of the coefficient matrix are cleared by transferring all known terms

to the right hand side, following the same procedure explained in (3.29). We thus arrive at

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 10 & 10 \\ 0 & 0 & 0 & 0 & 10 & 15 \end{bmatrix} \begin{bmatrix} u_{x1} \\ u_{x2} \\ u_{x2} \\ u_{y2} \\ u_{x3} \\ u_{y3} \end{bmatrix} = \begin{bmatrix} 0 \\ -0.5 \\ 0 \\ 0.4 \\ -3 \\ -2 \end{bmatrix} \tag{3.32}$$

As before, these are called the *modified master stiffness equations*. Note that (3.32) retains the original number and order as well as matrix symmetry. Solving this system yields the complete displacement solution (3.30).

If all prescribed displacements are zero, forces on the right hand side are not modified, and one would get (3.23) as may be expected.

Remark 3.5. The modification is not actually programmed as discussed above. First the applied forces in the right-hand side are modified for the effect of nonzero prescribed displacements, and the prescribed displacements stored in the reaction-force slots. This is called the *force modification* step. Second, rows and columns of the stiffness matrix are cleared as appropriate and ones stored in the diagonal positions. This is called the *stiffness modification* step. It is essential that the procedural steps be executed in the indicated order, because stiffness terms must be used to modify forces before they are zeroed out.

§3.6.3. *Matrix Forms of Nonzero-DBC Application Methods

The reduction and modification techniques for applying DBCs can be presented in compact matrix form. First, the free-free master stiffness equations $\mathbf{K}\mathbf{u} = \mathbf{f}$ are partitioned as follows:

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}. \tag{3.33}$$

In this matrix equation, subvectors $\mathbf{u}_2$ and $\mathbf{f}_1$ collect displacement and force components, respectively, that are *known*, *given* or *prescribed*. Subvectors $\mathbf{u}_1$ and $\mathbf{f}_2$ collect force and displacement components, respectively, that are *unknown*. Forces in $\mathbf{f}_2$ represent reactions on supports; consequently $\mathbf{f}_2$ is called the *reaction vector*.

On transferring the known terms to the right hand side the first matrix equation becomes

$$\mathbf{K}_{11}\mathbf{u}_1 = \mathbf{f}_1 - \mathbf{K}_{12}\mathbf{u}_2. \tag{3.34}$$

This is the *reduced master equation system*. If the support B.C.s are homogeneous (that is, all prescribed displacements are zero), $\mathbf{u}_2 = \mathbf{0}$, and we do not need to change the right-hand side:

$$\mathbf{K}_{11}\mathbf{u}_1 = \mathbf{f}_1. \tag{3.35}$$

Examples that illustrate (3.34) and (3.35) are (3.28) and (2.23), respectively.

The computer-oriented modification technique retains the same joint displacement vector as in (3.34) through the following rearrangement:

$$\begin{bmatrix} \mathbf{K}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 - \mathbf{K}_{12}\mathbf{u}_2 \\ \mathbf{u}_2 \end{bmatrix}. \tag{3.36}$$

This *modified system* is simply the reduced equation (3.35) augmented by the trivial equation $\mathbf{I}\mathbf{u}_2 = \mathbf{u}_2$. This system is often denoted as

$$\widehat{\mathbf{K}}\mathbf{u} = \hat{\mathbf{f}}. \tag{3.37}$$

Solving (3.37) yields the complete displacement solution including the specified displacements $\mathbf{u}_2$.

For the computer implementation it is important to note that the partitioned form (3.33) is only used to permit the use of compact matrix notation. In actual programming the equations are *not* explicitly rearranged: they retain their original numbers. For instance, in the example truss

$$\mathbf{u}_1 = \begin{bmatrix} u_{x1} \\ u_{y1} \\ u_{y2} \end{bmatrix} \equiv \begin{bmatrix} \text{DOF \#1} \\ \text{DOF \#2} \\ \text{DOF \#4} \end{bmatrix}, \qquad \mathbf{u}_2 = \begin{bmatrix} u_{x2} \\ u_{x3} \\ u_{y3} \end{bmatrix} \equiv \begin{bmatrix} \text{DOF \#3} \\ \text{DOF \#5} \\ \text{DOF \#6} \end{bmatrix}. \tag{3.38}$$

The example shows that $\mathbf{u}_1$ and $\mathbf{u}_2$ are generally interspersed throughout $\mathbf{u}$. Thus, matrix operations such as $\mathbf{K}_{12}\mathbf{u}_2$ involve indirect (pointer) addressing so as to avoid explicit array rearrangement.

Notes and Bibliography

The coverage of the assembly and solution steps of the DSM, along with globalization and application of BCs, is not uniform across the wide spectrum of FEM books. Authors have introduced "quirks" although the overall concepts are not affected. The most common variations arise in two contexts:

(1) Some treatments apply support conditions *during* merge, explicitly eliminating known displacement freedoms as the elements are processed and merged into $\mathbf{K}$. The output of the assembly process is what is called here a reduced stiffness matrix.[2]

(2) In the *frontal solution method* of Irons [102,103], assembly and solution are done concurrently. More precisely, as elements are formed and merged, displacement boundary conditions are applied, and Gauss elimination and reduction of the right hand side starts once the assembler senses (by tracking an "element wavefront") that no more elements contribute to a certain node.

Both variants appeared in FEM programs written during the 1960s and 1970s. They were motivated by computer resource limitations of the time: memory was scarce and computing time expensive.[3] On the negative side, interweaving leads to unmodular programming (which easily becomes "spaghetti code" in low-level languages such as Fortran). Since a frontal solver has to access the element library, which is typically the largest component of a general-purpose FEM program, it has to know how to pass and receive information about each element. A minor change deep down the element library can propagate and break the solver.

Squeezing storage and CPU savings on present computers is of less significance. Modularity, which simplifies scripting in higher order languages such as *Matlab* is desirable because it increases "plug-in" operational flexibility, allows the use of built-in solvers, and reduces the chance for errors. These changes reflect economic reality: human time is nowadays far more expensive than computer time.

A side benefit of modular assembly-solution separation is that often the master stiffness must be used in a different way than just solving $\mathbf{Ku} = \mathbf{f}$; for example in dynamics, vibration or stability analysis. Or as input to a model reduction process. In those cases the solution stage can wait.

Both the hand-oriented and computer-oriented application of boundary conditions have been presented here, although the latter is still considered an advanced topic. While hand computations become unfeasible beyond fairly trivial models, they are important from a instructional standpoint.

The augment-and-add procedure for hand assembly of the master stiffness matrix is due to H. Martin [116].

References

Referenced items have been moved to Appendix R.

---

[2] For the example truss, the coefficient matrix in (3.15) is a reduced stiffness whereas that in (3.23) is a modified one.

[3] As an illustration, the first computer used by the writer, the "classical mainframe" IBM 7094, had a magnetic-core memory of 32,768 36-bit words ($\approx$ 0.2 MB), and was as fast as an IBM PC of the mid 1980s. One mainframe had to serve the whole Berkeley campus, and Ph.D. students were allocated 2 CPU hours per semester. Getting a moderately complex FE model through involved heavy use of slower secondary storage such as disk or tape in batch jobs.