3 Sparseness

3.1 Algebraic manifestations of physical equilibrium

Many of the large systems of linear algebraic equations solved on present day computers arise from finite element and finite difference approximations of solid, fluid and other field *equilibrium and evolution* problems of computational mechanics. Replacement of a continuum equilibrium problem by a good algebraic one which correctly describes the detailed discrete behavior of an elastic solid, a moving fluid, a hot body, or magnetic field requires knowledge of the field values at very many points throughout the mass or space. Hence the sheer size of the algebraic system of equilibrium.

The nature of equilibrium profoundly affects the size, form and substance of the linear algebraic systems to which it gives rise, endowing equilibrium systems with distinctive characteristics that, to a large extent, set the agenda of computational linear algebra.

There are intimate connections between the physics of the algebraically approximated problem, even the most intuitive ones, and the deepest, most fundamental theoretical properties of the linear system of equations which formulate the equilibrium; we cannot understand the gist and purpose of the mathematics without a good understanding of the physics.

In this chapter we will consider basic linear algebraic issues particular to equilibrium problems. Because the nature of equilibrium is so central to the discussion we shall be careful to derive the mathematics of equilibrium for some simple yet typical continuum equilibrium problems from their very underlying physical principles.

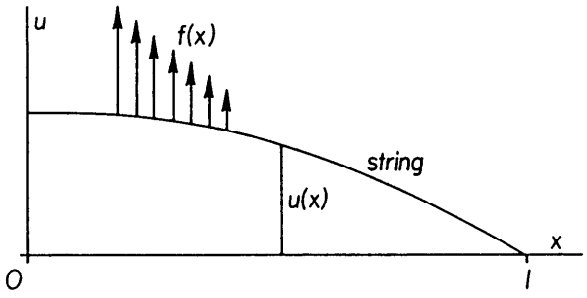
Discretization— the passage from the continuous to the discrete, is accomplished here by means of finite difference approximations. Discussion of the more sophisticated finite element method and its linear algebraic ramifications is deferred to the last chapter of this book.

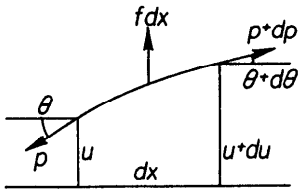
3.2 Finite differences–the taut string

Herewith we shall consider the conception of discretization –of approximating a continuous process by a discrete algebraic one. The eminent example of the equilibrium of a laterally forced taut string is simple enough for a detailed mathematical examination, yet realistic enough to explicitly disclose the parallels between the physics and the algebra.

As is common in the analysis of such problems, we shall first write down the *differential* equation of equilibrium for the string, and then approximate the equation by finite differences.

What we mean by *string* is a thin, long piece of *elastic* solid able to carry *tensional axial* forces only. Such loaded string is shown in Fig.3.1(a). It is under axial tension $p > 0$, is acted upon by a lateral distributed force $f(x)$, is fixed at point $x = 1$, and is symmetric about the u axis. In response to the action of the applied distributed force $f(x)$ the string deflects and stretches, but we shall assume that the lateral displacement $u(x)$ is very small in magnitude compared with the original length of the string, $|u(x)| << 1$, and that the slope of the deflection is of a magnitude much smaller than unity, $|u'(x)| << 1$, implying essentially lateral loadings that are very small compared with the axial tension.





(a) Fig. 3.1 (b)

Tension, the force that one part of the string exerts on another, is caused by either an

initial stretch, inertia, gravity, magnetic body forces, or an excessive lateral deformation. In principle tension can be a function of both x and $u(x)$.

To write the differential equations of equilibrium for the string, we reckon the vertical and horizontal forces that act on a differential segment dx of it as shown in Fig. 3.1(b). The horizontal and vertical zero force sums are expressed, in the absence of an external axial pull, as

$$-p\sin\theta + (p+dp)\sin(\theta+d\theta) + f(x)dx = 0 \tag{3.1}$$

and

$$-p\cos\theta + (p+dp)\cos(\theta+d\theta) = 0 \tag{3.2}$$

respectively. But

$$\sin(\theta+d\theta) = \sin\theta + \cos\theta d\theta,\ \cos(\theta+d\theta) = \cos\theta - \sin\theta d\theta \tag{3.3}$$

since $\cos(d\theta) = 1, \sin(d\theta) = d\theta$, and the two equations of equilibrium reduce to

$$d(p\sin\theta) + f(x)dx = 0,\ \text{and}\ \ d(p\cos\theta) = 0. \tag{3.4}$$

Integration of the second of eqs.(3.4) produces $p\cos\theta = p_0$, and if θ is small so that $\cos\theta = 1$, then $p = p_0$ independently of θ. The assumption of small displacements *decouples* displacement $u(x)$ from tension $p(x)$. Generally, tension $p(x)$ is computed first from the actions of the external forces, then inserted as a given coefficient in the equation of vertical equilibrium.

Thus, with $\sin\theta = \theta = u' = du/dx$, equilibrium of the string is described by the differential equation

$$(pu')' + f(x) = 0 \quad 0 < x < 1 \tag{3.5}$$

with $p = p(x)$ given.

Equation of equilibrium (3.5) is supplemented by the two *boundary conditions*

$$u(1) = u'(0) = 0 \tag{3.6}$$

at end points $x = 0$ and $x = 1$.

Equations (3.5) and (3.6) constitute a *two-point boundary value problem* of the string. Understandably, without boundary condition $u(1) = 0$ that holds down the string, the

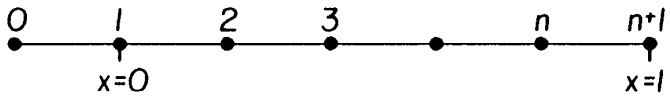
boundary value problem would have possessed many solutions of the form $u(x) + c$ for arbitrary constant c, and for u satisfying equilibrium equation (3.5), boundary condition $u'(0) = 0$, and possibly $u'(1) = 0$.

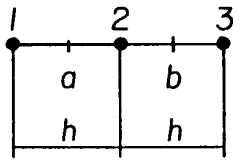
For the rest of this discussion we shall conveniently suppose constant unit tension, $p = 1$, so as to have the simpler

$$\begin{gathered} -u'' = f(x) \quad 0 < x < 1 \\ u'(0) = u(1) = 0 \end{gathered} \tag{3.7}$$

in place of eqs. (3.5) and (3.6).

To discretize the string, that is, to replace its analytic differential equilibrium formulation (3.7) by an approximate linear algebraic one, we divide the string into n equal segments of length $h = 1/n$, as in Fig. 3.2(a), with intermediate *nodes* labeled $1, 2, \ldots, n, n+1$ to which we assign all string data. Fictitious node 0 is added under the assumption of a symmetric continuation of the string beyond $x = 0$, and is placed there for the purpose of helping in the approximation of the boundary conditions.





(a) Fig. 3.2 (b)

Algebra is instituted for analysis, and the discrete for the continuous, by replacing the differential equation of equilibrium, good for any point along the string, by an algebraic system of equilibrium equations written at interior nodes only and involving nodal values only. Boundary conditions are added likewise.

A finite difference approximation to u_2'', u'' at node 2 of Fig. 3.2(b), is written with the repeated approximations

$$u_a' = \frac{1}{h}(u_2 - u_1), u_b' = \frac{1}{h}(u_3 - u_2), u_2'' = \frac{1}{h}(u_b' - u_a') \tag{3.8}$$

as

$$u_2'' = \frac{1}{h^2}(u_1 - 2u_2 + u_3) \tag{3.9}$$

where subscripts refer to the node numbers and where prime stands for differentiation with respect to x. With eq. (3.9) differential equation (3.7) is approximated by:

$$\begin{aligned}
&\text{at node } 2 \quad \frac{1}{h^2}(-u_1 + 2u_2 - u_3) = f_2, \\
&\text{at node } 3 \quad \frac{1}{h^2}(-u_2 + 2u_3 - u_4) = f_3, \\
&\qquad\qquad\qquad \vdots \\
&\text{at node } n \quad \frac{1}{h^2}(-u_{n-1} + 2u_n - u_{n+1}) = f_n.
\end{aligned} \tag{3.10}$$

At the last node boundary condition $u_{n+1} = 0$ prevails, but we still need to approximate $u'(0) = 0$. Making use of fictitious node 0 we write the approximations

$$u_1' = \frac{1}{2h}(u_2 - u_0) = 0 \ , \ \frac{1}{h^2}(u_0 - 2u_1 + u_2) = f_1 \tag{3.11}$$

and upon the elimination of u_0 between them are left with

$$\frac{1}{h^2}(u_1 - u_2) = \frac{1}{2}f_1 \tag{3.12}$$

at point 1.

Equations (3.10) together with $u_{n+1} = 0$ and eq.(3.12) are collected in the linear system

$$\frac{1}{h}\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ \vdots \\ \\ u_n \end{bmatrix} = h \begin{bmatrix} \frac{1}{2}f_1 \\ f_2 \\ f_3 \\ \vdots \\ \\ f_n \end{bmatrix} \ , \ Ku = f \tag{3.13}$$

for the unknown nodal displacements vector u. The discrete counterpart to the linear two-point boundary value problem (3.7) is a system of linear algebraic equations expressing approximate equilibrium at the nodes. In system (3.13), K is the *stiffness* matrix and f the *load* vector. We notice that the load vector consists of point forces averaged over the interval h around each node. At node 1 the force is only $1/2 f_1$ as the other half of the force is lost to symmetry.

We observe that stiffness matrix K is:

1. *Symmetric.*

2. *Sparse*, with many zero entries.

3. Of a *band* form with the nonzero entries close and parallel to the main diagonal.

4. *Tridiagonal*.

5. With *repetitive entries*.

Symmetry in K stems from the string deflection being described by an *even degree* differential equation, from the *central* or symmetric finite difference formula for u'', and from boundary conditions that are just right. Boundary value problems that produce symmetric finite difference systems are *self-adjoint* and constitute the most interesting class of problems in computational mechanics.

Sparseness and the band form of K stems from the differential equation that expresses equilibrium *at a point*, and from the *consecutive* node numbering. The nodal discrete finite difference equations of equilibrium involve only neighboring nodes. Band form is inherent in equilibrium problems and we introduce the

Definition. *Square matrix K is a band matrix of bandwidth $2k+1$ if for $|i-j| > k, K_{ij} = 0$. By band (K) we designate all entries K_{ij} such that $|i-j| \leq k$.*

A tridiagonal matrix, for instance, is with $k = 1$.

The critical question of the nonsingularity of K in eq. (3.13) is resolved by the LL^T factorization

$$hK = LL^T, \quad L = \begin{bmatrix} 1 & & & & & & & \\ -1 & 1 & & & & & & \\ & -1 & 1 & & & & & \\ & & -1 & 1 & & & & \\ & & & -1 & 1 & & & \\ & & & & -1 & 1 & & \\ & & & & & -1 & 1 & \\ & & & & & & -1 & 1 \end{bmatrix} \tag{3.14}$$

demonstrating that K is not only nonsingular but also positive definite with equal unit pivots. In light of eq.(3.14), factorization of K into LL^T appears to be the discrete counterpart to the factorization of the *second-order differential operator* of the string problem into two first-order differential operators.

If boundary condition $u'(1) = 0$ prevails at $x = 1$,instead of $u(1) = 0$, then the *homoge-*

neous two-point boundary value problem

$$u'' = 0 \quad 0 < x < 1, u'(0) = u'(1) = 0 \tag{3.15}$$

is solved by the nontrivial $u = \ c \ \neq 0$ for arbitrary constant c, which obviously satisfies both the equation of equilibrium and two boundary conditions. Corresponding to problem (3.15) is the stiffness matrix

$$K = \frac{1}{h}\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 \end{bmatrix} \tag{3.16}$$

verified to be singular by $Ku = o, \ u = [1 \ 1 \ \ldots \ 1]^T$.

As a lower-triangular factor in $hK = LL^T$ we compute for K in eq. (3.16)

$$L = \begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 & \\ & & & & -1 & 0 \end{bmatrix} \tag{3.17}$$

and the zero pivot is encountered last since the singularity or nonsingularity of K is decided only at the last equation, which expresses the second boundary condition.

A string with both ends fixed, with $u(1) = u(0) = 0$, gives rise to the stiffness matrix

$$K = \frac{1}{h}\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \tag{3.18}$$

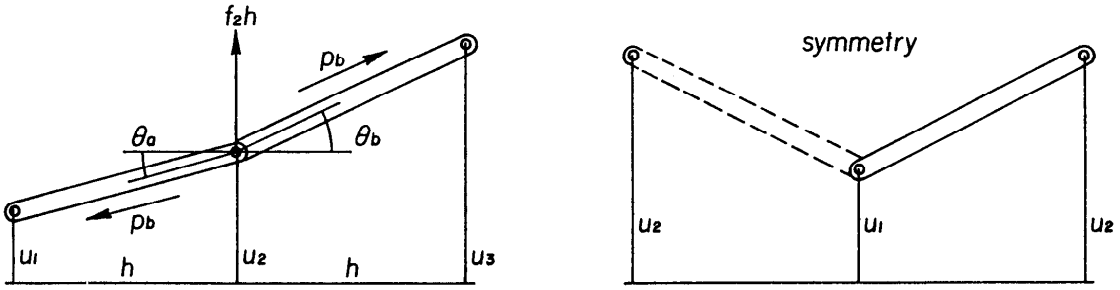
and in $K = LDL^T$

$$L = \begin{bmatrix} 1 & & & & & \\ -\frac{1}{2} & 1 & & & & \\ & -\frac{2}{3} & 1 & & & \\ & & -\frac{3}{4} & 1 & & \\ & & & -\frac{4}{5} & 1 & \\ & & & & -\frac{5}{6} & 1 \end{bmatrix}, D = \frac{1}{h}\begin{bmatrix} \frac{2}{1} & & & & & \\ & \frac{3}{2} & & & & \\ & & \frac{4}{3} & & & \\ & & & \frac{5}{4} & & \\ & & & & \frac{6}{5} & \\ & & & & & \frac{7}{6} \end{bmatrix}. \tag{3.19}$$

Pivots $D_{ii} = (1 + 1/i)/h$ are not all equal, but pivoting with this positive definite matrix is certainly not necessary.

3.3 Elastic energy

The method of initially writing down the differential equation of equilibrium and boundary conditions and then approximating them by finite differences has its mathematical merits as we shall see in the next two sections, but we can also write the nodal equations of equilibrium directly from a *discrete mechanical model*. Since the string is elastic but transmits only tential axial forces we imagine it as consisting of a linkage of short, thin elastic ties connected by means of frictionless pins, as in Fig.3.3.



(a) Fig. 3.3 (b)

For finite differences the string exists at the nodes only, but the mechanical model gives internodal substance to the discretization. Distributed forces are equivalently apportioned, or *lumped*, at the joints, and the equation of equilibrium in the vertical direction for typical joint 2 of Fig. 3.3(a) is written as

$$h f_2 - p_a \sin\theta_a + p_b \sin\theta_b = 0. \tag{3.20}$$

Under the assumption of small displacements

$$\sin\theta_a = \frac{1}{h}(u_2 - u_1), \sin\theta_b = \frac{1}{h}(u_3 - u_2) \tag{3.21}$$

and joint 2 is at equilibrium on condition that

$$h^2 f_2 + u_1 p_a - u_2(p_a + p_b) + u_3 p_b = 0 \tag{3.22}$$

reverting to $f_2 + (u_1 - 2u_2 + u_3)/h^2 = 0$ if $p_a = p_b = 1$. For joint 1 on the line of symmetry we have with reference to Fig. 3.3(b) that

$$f_1 h + 2p_a \sin\theta_1 = 0 \ , \ \sin\theta_1 = \frac{1}{h}(u_2 - u_1) \tag{3.23}$$

which with $p_a = 1$ becomes the first equation in system (3.13).

The LDL factorization proves that stiffness matrix K is positive definite and hence nonsingular, provided that the string is fixed at least at one of its end points. Positive definiteness can be directly shown for K in eq. (3.13) by the expansion

$$\begin{aligned} hu^T Ku &= u_1^2 + 2u_2^2 + 2u_3^2 + \ldots + 2u_n^2 \\ &\quad - 2u_1u_2 - 2u_2u_3 - \ldots - 2u_{n-1}u_n \\ &= (u_2 - u_1)^2 + (u_3 - u_2)^2 + \ldots + (u_n - n_{n-1})^2 + (u_{n+1} - u_n)^2, u_{n+1} = 0 \end{aligned} \tag{3.24}$$

demonstrating that $u^T Ku > 0$ if $u \neq o$; only when $u = o$ is $u^T Ku = 0$.

Notice that quadratic form $hu^T Ku$ is for arbitrary u, not just for u satisfying $Ku = f$, for which $u^T Ku = f^T K^{-1} f$. Nevertheless, stiffness matrix K includes boundary condition $u_{n+1} = 0$, without which K is only positive semidefinite. Boundary condition $u'(0) = 0$ as expressed in eqs. (3.11) and (3.12) includes f_1, and an arbitrary u disregards this condition. But it matters little what conditions are imposed on u at $x = 0$; K is positive definite in any event. Positive definiteness in K expresses a deep and fundamental physical property of the string deformation. Quadratic form $\frac{1}{2}u^T Ku$ is physically interpreted as the *elastic energy* stored in the string by deformation u, or geometrically, for unit tension, as the total *elongation* suffered by the string during deflection. Any lateral change of form causes the string to stretch and elongate and therefore only increases the level of stored elastic energy.

What is the elastic energy stored in the analytically modeled string? Recall that the string deflections and rotations are all small and that, as a result, the string tension is independent of its displacement. Figure 3.4 shows a differential segment dx of the string elongated by a small lateral deflection.

The elastic energy stored in the string element equals the work of tension $p = p(x)$ exerted to extend it from length dx to length $(1 + \epsilon)dx$, and is equal to $p\epsilon dx$, where $\epsilon = \epsilon(x)$

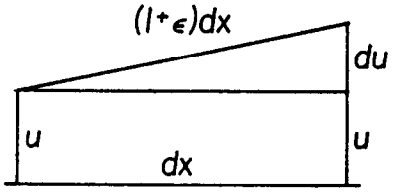


Fig. 3.4

in the pointwise relative string extension or *strain*. We have for the right differential triangle in Fig. 3.4 that

$$1 + \epsilon = \sqrt{1 + u'^2} \tag{3.25}$$

and if $|u'(x)| << 1$, then

$$\epsilon = \frac{1}{2}u'^2 \tag{3.26}$$

and the total elastic energy stored in the entire string of length 1 becomes

$$\mathcal{E} = \frac{1}{2}\int_0^1 pu'^2 dx. \tag{3.27}$$

For the rod linkage model we write $dx = h, du = u_2 - u_1$, and have for one tie

$$p\epsilon h = \frac{1}{2}\frac{p}{h}(u_2 - u_1)^2 \tag{3.28}$$

which, with $p = 1$, is proportional to a typical term in eq.(3.24). If we write the string model tie element displacements as vector $u = [u_1 \ u_2]^T$, then quadratic form (3.28) assumes the linear algebraic form

$$p\epsilon h = \frac{1}{2}u^T k u \ , \ k = \frac{p}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{3.29}$$

in which k is the *element stiffness* matrix of one link. It is customary to denote this small size matrix by lower-case k, and we adhere to this convention.

Elastic energy expressions (3.28) and (3.29) amount to a piecewise integration of $\mathcal{E}$ in eq. (3.27) under the tacit assumption that pu' is constant in the interval between two nodes.

3.4 Greater accuracy

Accuracy of the finite difference approximation of boundary value problems can be improved in two ways: 1. by subdividing the string into smaller segments, and 2. by using more accurate finite difference formulas.

To understand how more accurate finite difference schemes are devised reconsider

$$u_1'' = \frac{1}{h^2}(u_0 - 2u_1 + u_2) \tag{3.30}$$

at $x_0 = -h, x_1 = 0, x_2 = h$, and notice that it *correctly* computes u'' for $u = 1, u = x, u = x^2$ in the *entire* interval between points 0 and 2. For $u = x^3$, u'' is correctly computed by eq. (3.30) at *central* node 1 only, where $u'' = 0$. For $u = x^4$ eq. (3.30) yields $u_1'' = 2h^2$, which becomes ever smaller as $h \to 0$, but is nonetheless inaccurate. We want the finite difference formula to be accurate, or consistent, for a polynomial u of as high a degree as possible since according to Taylor's theorem, a function that is analytic at x is nearly polynomial if consideration of it is confined to a sufficiently small interval around x.

To have a finite difference scheme that correctly computes u'' for a quartic u we must include more points in the formula. In fact, the approximation

$$u_2'' = \frac{1}{12h^2}(-u_0 + 16u_1 - 30u_2 + 16u_3 - u_4) \tag{3.31}$$

does it. It correctly computes u_2'' for $u = 1, u = x, u = x^2, u = x^3, u = x^4$ and $u = x^5$. For $u = x^6$ formula (3.31) bears $u_2'' = -8h^4$, which is very small in magnitude compared to 1 if h is much smaller than 1.

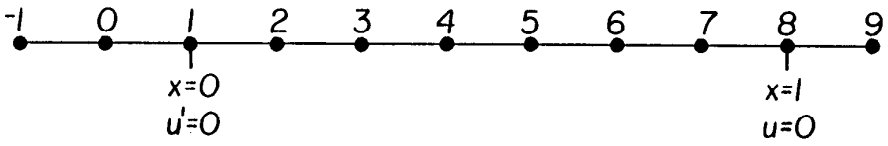


Fig. 3.5

Approximation formula (3.31) for u'' includes five neighboring points, and we need more outside fictitious nodes to approximate the equation and boundary conditions of $u'' + f = 0$ $0 < x < 1, u'(0) = u(1) = 0$. In Fig. 3.5 the string extends between nodes 1 and 8, points labeled $-1, 0, 9$ being fictitious. Points -1 and 0 are justified by symmetry, but for point 9 we must assume a polynomial extension of the string behind the right-hand fixing support. We assign to the nodes concentrated loads $f_1, f_2, \ldots, f_8$, and going from node to node write:

$$
\begin{aligned}
&\text{at node 1} \quad \frac{1}{12h^2}(u_{-1} - 16u_0 + 30u_1 - 16u_2 + u_3) = f_1,\\
&\text{at node 2} \quad \frac{1}{12h^2}(u_0 - 16u_1 + 30u_2 - 16u_3 + u_4) = f_2,\\
&\text{at node 3} \quad \frac{1}{12h^2}(u_1 - 16u_2 + 30u_3 - 16u_4 + u_5) = f_3,
\end{aligned}
\tag{3.32}
$$

then we eliminate u_{-1} and u_0 from among them using the symmetry conditions $u_0 = u_2, u_{-1} = u_3$.

At the other end of the string we write:

$$
\begin{aligned}
&\text{at node 6} \quad \frac{1}{12h^2}(u_4 - 16u_5 + 30u_6 - 16u_7 + u_8) = f_6,\\
&\text{at node 7} \quad \frac{1}{12h^2}(u_5 - 16u_6 + 30u_7 - 16u_8 + u_9) = f_7,
\end{aligned}
\tag{3.33}
$$

and set in these equations $u_8 = 0$. Then we write the less accurate

$$
\frac{1}{h^2}(-u_7 + 2u_8 - u_9) = f_8 \tag{3.34}
$$

and use it to eliminate u_9 from eq.(3.33). We need not be concerned about the lower accuracy of eq (3.34) because around point 8 the displacements are the smallest. A better finite difference approximation to u'' calls for a better approximation to f, but since here we are mainly interested in the stiffness matrix rather than the load vector we leave this issue to the exercises.

In matrix form equations (3.32), (3.33) and (3.34) become $Ku = f$,

$$
\frac{1}{6h}\begin{bmatrix}
15 & -16 & 1 & & & & \\
-16 & 31 & -16 & 1 & & & \\
1 & -16 & 30 & -16 & 1 & & \\
 & 1 & -16 & 30 & -16 & 1 & \\
 & & 1 & -16 & 30 & -16 & 1 \\
 & & & 1 & -16 & 30 & -16 \\
 & & & & 1 & -16 & 29
\end{bmatrix}
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \end{bmatrix}
= 2h \begin{bmatrix} \frac{1}{2}f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 + \frac{1}{12}f_8 \end{bmatrix}. \tag{3.35}
$$

Devising a mechanical model for the higher-order finite difference scheme is not obvious anymore, and herein lies the advantage of the purely mathematical approach to discretization that starts from a differential equation and approximately replaces it by finite differences.

We verify for eq. (3.35) that

$$\begin{aligned} 6h\ u^T K u &= 8(u_2-u_1)^2 \\ &+ \sum_{j=1}^{n-1}\Big(6(u_j-u_{j+1})^2+6(u_{j+1}-u_{j+2})^2+(u_j-2u_{j+1}+u_{j+2})^2\Big) \\ &+ 6u_n^2\ ,\ u_{n+1}=0 \end{aligned} \tag{3.36}$$

for any u, and K is positive definite.

Also, for the higher-order finite difference approximations, $\frac{1}{2}u^T K u$ still means energy and elongation. In the finite difference modeling, the string and its properties exist only at the nodes. But let us suppose that the string displacement is interpolated parabolically between any three nodes, say 1,2 and 3. Then in the interval between nodes 1 and 3 the deflection is written as

$$u = u(x) = u_1\frac{1}{2}\xi(\xi-1)+u_2(1-\xi^2)+u_3\frac{1}{2}\xi(\xi+1),\ -1\le\xi\le 1 \tag{3.37}$$

where $x = x_2 + h\xi$. The small deflection elongation, or elastic energy, of the parabolic segment is given by

$$\begin{aligned} \frac{1}{2}\int_{-h}^{h} u'^2 dx &= \frac{1}{2h}\int_{-1}^{1}\dot{u}^2 d\xi \\ &= \frac{1}{12h}(7u_1^2+16u_2^2+7u_3^2-16u_1u_2-16u_2u_3+2u_1u_3) \end{aligned} \tag{3.38}$$

where $\dot{u} = du/d\xi$. With $u = [u_1\ u_2\ u_3]^T$ quadratic form (3.39) is linear algebraically expressed as

$$\frac{1}{2}\int_{-h}^{h} u'^2 dx = \frac{1}{2}u^T k u\ ,\ k = \frac{1}{6h}\begin{bmatrix} 7 & -8 & 1 \\ -8 & 16 & -8 \\ 1 & -8 & 7 \end{bmatrix} \tag{3.39}$$

in which k is the stiffness matrix of the parabolic string element.

Comparing equations (3.38) and (3.36) we realize that $\frac{1}{2}u^T K u$ may be interpreted as the elastic energy of the entire string considered made of *overlapping* parabolic segments as in Fig. 3.6, in which $u_0 = u_2, u_8 = 0$, where only half the energy of the extreme segments is added, and where the last segment is linear, $u_9 = -u_7$.

The point is this: *A finer segmentation of the string done in order to achieve a better finite difference approximation creates larger linear systems with more unknowns. Higher-order finite difference schemes increase the bandwidth of matrix K*, which in eq.(3.35) is 5 instead of 3 in eq.(3.13).

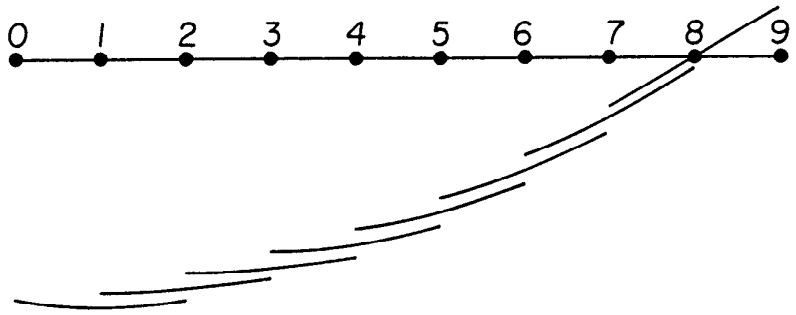


Fig. 3.6

Another noticeable difference between the low- and high-order stiffness matrices: for K in eq. (3.13) the absolute sum of the off-diagonal entries in each row is not larger than the corresponding diagonal entry. Such a case is called *diagonal dominance*. It is no longer true for K in eq.(3.35).

Because we want to retain clear physical significance in the linear algebraic analysis of the string problem, we shall return to the simpler lower-order discrete model of eq.(3.13).

Exercises

3.4.1. Let $u = [u_1 \ u_2 \ \dots \ u_n]^T$ include the *exact* nodal values of the string problem as obtained from the solution of boundary value problem (3.7). This vector does not satisfy algebraic system (3.13) exactly but leaves a residual vector r. For typical interior node j

$$\frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}) + f_j = r_j.$$

Expand u_{j+1} and u_{j-1} by means of Taylor's theorem around point j, and show that r_j, including r_1, is proportional to h^2.

3.4.2. Let u be the exact nodal values vector and u' the approximate nodal values vector, $f - Ku' = o$, $f - Ku = r$, so that $K(u - u') = r$ and $u - u' = K^{-1}r$. Show that in discretization (3.13), $\|u - u'\|_\infty$ is proportional to h^2.

3.4.3. A finite difference scheme of a higher degree of consistency is obtained for the string by the inclusion of more f nodal values. Instead of approximation (3.10) write

$$\frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}) + \alpha f_{j-1} + \beta f_j + \alpha f_{j+1} = r_j$$

and, assuming exact nodal values, use Taylor's theorem to determine α and β so that r_j is proportional to h^p with highest p. Do the same for approximation (3.31).

3.4.4. Suppose that point j is a point of discontinuity for load $f(x)$. Repeat exercise 3.4.3 for this case and write the best approximate force for the discrete equation of equilibrium at node j in terms of f_{j-1}, f_j^-, f_j^+ and f_{j+1}. Is the force of exercise 3.4.3 recovered with $f_j^- = f_j^+ = f_j$?

3.4.5. Use finite differences to algebraize the string problem

$$u'' + f(x) = 0 \qquad 0 < x < 1$$
$$u(0) = u(1) = 0$$

with

$$f(x) = 1 \ \ \frac{1}{4} \le x \le \frac{3}{4} \ \ , \ f(x) = 0 \ \ \text{otherwise,}$$

and compare the approximate solution to the exact.

3.4.6. Discretize the overdetermined boundary value problem

$$u'' + f(x) = 0,\ 0 < x < 1,\ f(0) = f(1) = 0.$$
$$u(0) = 0 \quad u(1) = 0$$
$$u'(0) = 0 \quad u'(1) = 0$$

by finite differences and discuss what happens to the solution as $h \to 0$.

3.5 The flexibility matrix

In the terminology of computational mechanics, matrix K of $Ku = f$ is the *stiffness* matrix of the string problem. Its inverse $F = K^{-1}$ is the *flexibility* matrix. We want to explore here some of the basic properties of F.

For the stiffness matrix of the symmetric string,

$$\frac{1}{h}\begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}^{-1} = h\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 & 1 \\ & & 1 & 1 & 1 & 1 \\ & & & 1 & 1 & 1 \\ & & & & 1 & 1 \\ & & & & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & & & \\ 1 & 1 & & & & \\ 1 & 1 & 1 & & & \\ 1 & 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

$$= h \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 5 & 4 & 3 & 2 & 1 \\ 4 & 4 & 4 & 3 & 2 & 1 \\ 3 & 3 & 3 & 3 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, h = \frac{1}{n}, \tag{3.40}$$

whereas for the fixed string,

$$\frac{1}{h} \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & 2 & -1 & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}^{-1} = \frac{h}{n+1} \begin{bmatrix} 6 & 5 & 4 & 3 & 2 & 1 \\ 5 & 10 & 8 & 6 & 4 & 2 \\ 4 & 8 & 12 & 9 & 6 & 3 \\ 3 & 6 & 9 & 12 & 8 & 4 \\ 2 & 4 & 6 & 8 & 10 & 5 \\ 1 & 2 & 3 & 4 & 5 & 6 \end{bmatrix}, h = \frac{1}{n+1}. \tag{3.41}$$

Analytically,

$$F_{ij} = \frac{1}{n}(n+1-j), j \geq i \tag{3.42}$$

for $F = K^{-1}$ in eq.(3.40), whereas

$$F_{ij} = \frac{1}{(n+1)^2} i(n+1-j), j \geq i \tag{3.43}$$

for F in eq.(3.41).

We introduce the measure, or norm

$$\|K\|_\infty = \max_i \sum_j |K_{ij}| \tag{3.44}$$

of K, and compute for K in eq. (3.13) and the corresponding F in eq. (3.40),

$$\|K\|_\infty = 4/h \text{ and } \|K^{-1}\|_\infty = h(1+n)n/2 \tag{3.45}$$

so that when n is large

$$\kappa_\infty = \|K\|_\infty \|K^{-1}\|_\infty = 2n^2. \tag{3.46}$$

Notice that for a positive $K^{-1} = F$, $F_{ij} > 0$, $\|F\|_\infty$ equals the maximum u_i in $Ku = e$, $e = [1\ 1\ \ldots\ 1]^T$.

Flexibility matrix F is symmetric and positive definite since stiffness matrix K is such. We also observe that F is:

1. *Dense*, not one of its entries is zero.

2. *Positive*, $F_{ij} > 0$ for all i and j.

3. *Bounded*, $\max F_{ij} = 1$ in eq.(3.40), and $\max F_{ij} = 1/4$ in eq.(3.41) if n is odd, independently of n.

For a physical interpretation of these three important observations notice, as in Fig. 3.7, that the jth column of flexibility matrix F contains the displacements due to a *single concentrated unit force* applied to the jth node. If

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} & F_{13} & F_{14} \\ F_{12} & F_{22} & F_{23} & F_{24} \\ F_{13} & F_{23} & F_{33} & F_{34} \\ F_{14} & F_{24} & F_{34} & F_{44} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \text{ then } \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} F_{13} \\ F_{23} \\ F_{33} \\ F_{34} \end{bmatrix}. \tag{3.47}$$

It is in the physical nature of the string that a concentrated force applied at any interior point causes *all* points of the string to move, and all *in the direction* of the applied force. The finite difference deflection computed for the point-loaded string is theoretically *exact* for any number of nodes, and hence the maximum deflection, and with it $\max F_{ij}$, is fixed for any n.

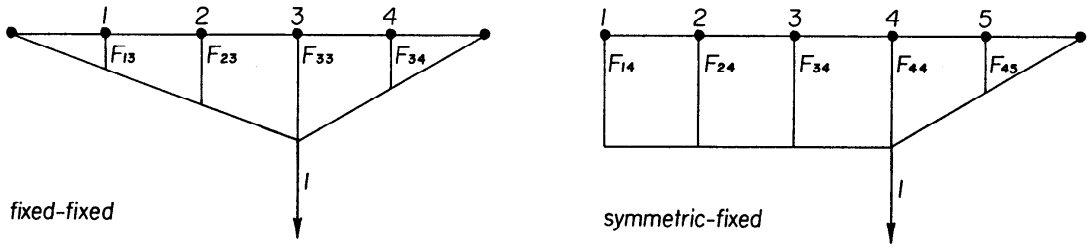


Fig. 3.7

A string cannot transmit rotation or torque (one cannot use a string or rope as a lever), and under a concentrated load it abruptly changes slope. Consequently if the string is *internally* fixed, then the problem of computing its displacements separates, as in Fig. 3.8, into two disjoint problems between the two pairs of supports.

Symmetry in F constitutes a discrete counterpart to the *Betti-Maxwell* reciprocal theorem of elasticity: *A unit force at node i causes the same deflection at node j as a unit force at j causes at i.* See Fig. 3.9.

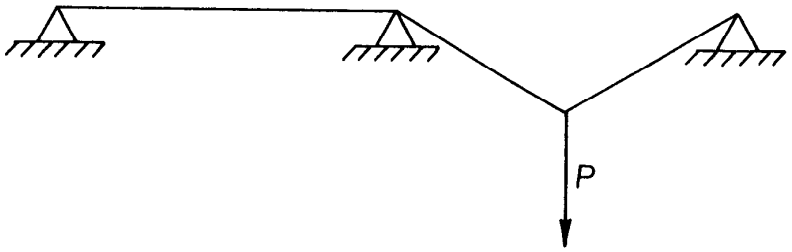


Fig. 3.8

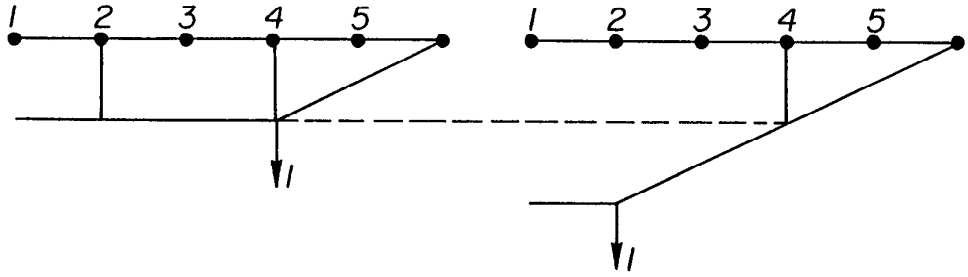


Fig. 3.9

An *experimental* F is constructed by measuring the nodal deflections resulting from a unit force applied sequentially at all nodes. Because F is symmetric the deflection curve due to one force can be determined by measuring u at a *fixed* point, where it may be measured most conveniently and accurately, while shifting the *force* from point to point.

To this extent the algebraic formulation of the string faithfully imitated the analytical model of the string; both sensibly duplicating nature. It must be borne in mind, however, that in general the analytical or physical properties of the algebraically described problem may correctly appear only in the limit of the discretization. Physical intuition can be a revealing guide, but is no substitute to proper mathematical deliberation.

Exercises

3.5.1. Write symmetric flexibility matrix F of eq.(3.41) as $F = (1+n)^{-2}M$, and show that $M_{ij} = x_i y_j \; j \geq i$, where x_i is the ith component of $x = [1\ 2\ 3\ 4\ 5\ 6]^T$, and y_j is the jth component of $y = [6\ 5\ 4\ 3\ 2\ 1]^T$. Give this a physical explanation.

3.6 Variable coefficients

Tension that is not constant along the string typifies variable coefficients in the differential equation of equilibrium. Linear algebraic issues that arise from variable coefficients are illustrated on the hanging string pulled down by its own weight.

At the lower end of the string tension is zero, and in a uniform string (think of a chain with very fine links), the tension grows linearly with the distance from the loose tip. Simply, $p(x) = x$. Vanishing tension at the lower end of the hanging string makes consideration of equilibrium more delicate at this point . The behavior of the loose end will require our attention but for the time being, to keep matters simple, we assume symmetry at $x = 0, u'(0) = 0$, and have

$$\begin{gathered} -(xu')' = f \qquad 0 < x < 1 \\ u'(0) = u(1) = 0 \end{gathered} \tag{3.48}$$

as the equation of equilibrium and boundary conditions for the laterally forced (say by wind) hanging string.

When a string is truly fixed against axial as well as lateral movement at both ends as in a violin or piano, its deflection is necessarily elastic. But we may also think of the tension as being supplied by a weight W transmitted over a pulley support as in Fig. 3.10. In this case the string can be assumed *inextensional* and the energy it stores in deflection becomes that of the weight lifted by the sagging, and

$$\mathcal{E} = \frac{1}{2} W \int_0^1 {u'}^2 dx \tag{3.49}$$

as for the inextensional string.

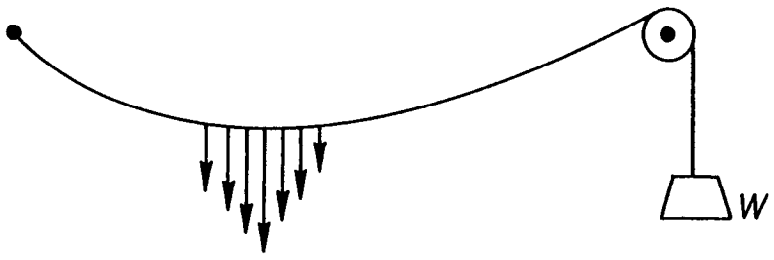


Fig. 3.10

A string with a free end can certainly be assumed inextensional, and for the hanging string the energy gained in deflection is the potential energy of its own weight raised by its own curving. The energy is discretely given by the quadratic form $\frac{1}{2}u^T Ku$ and it is worthwhile to have a theoretical expression for it. Refer to Fig. 3.11(a). A differential element dy of the string weights $\rho g dy$ and is lifted by deflection to gain energy by the amount

$$d\mathcal{E} = \frac{1}{2}\rho g dy \int_0^y u'^2(t)dt. \tag{3.50}$$

We assume $\rho g = 1$ and have that

$$\mathcal{E} = \frac{1}{2}\int_{y=0}^1 \int_{t=0}^y u'^2(t)dtdy. \tag{3.51}$$

Or with the order of integration reversed

$$\mathcal{E} = \frac{1}{2}\int_{t=0}^1 \int_{y=t}^1 u'^2(t)dydt \tag{3.52}$$

so that

$$\mathcal{E} = \frac{1}{2}\int_0^1 u'^2(1-t)dt \tag{3.53}$$

which with $x = 1 - t$ becomes

$$\mathcal{E} = \frac{1}{2}\int_0^1 xu'^2 dx \tag{3.54}$$

as for the elastic string.

If node 1 is placed at the lower end of the string, then the finite difference equation of equilibrium for the jth node is

$$-(xu')'_j = \frac{1}{2h}\left(-(2j-3)u_{j-1} + 4(j-1)u_j - (2j-1)u_{j+1}\right) = f_j \quad j = 2, 3, \ldots, n \tag{3.55}$$

with $u_{n+1} = 0$ to account for the fixed top. At node 1, due to symmetry

$$\frac{1}{2h}(-u_0 + 2u_1 - u_2) = f_1,\ u_0 = u_2 \tag{3.56}$$

and the linear system for the n node discretization of the hanging string becomes

$$\frac{1}{2}\begin{bmatrix} 1 & -1 & & & \\ -1 & 4 & -3 & & \\ & -3 & 8 & -5 & \\ & & -5 & 12 & -7 \\ & & & -7 & 16 \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \end{bmatrix} = h\begin{bmatrix} \frac{1}{2}f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix},\ Ku = f \tag{3.57}$$

with a stiffness matrix that factors into

$$2K = \begin{bmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & -1 & 1 & & & \\ & & -1 & 1 & & \\ & & & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & & & & \\ & 3 & & & \\ & & 5 & & \\ & & & 7 & \\ & & & & 9 \end{bmatrix} \begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & -1 & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix}. \tag{3.58}$$

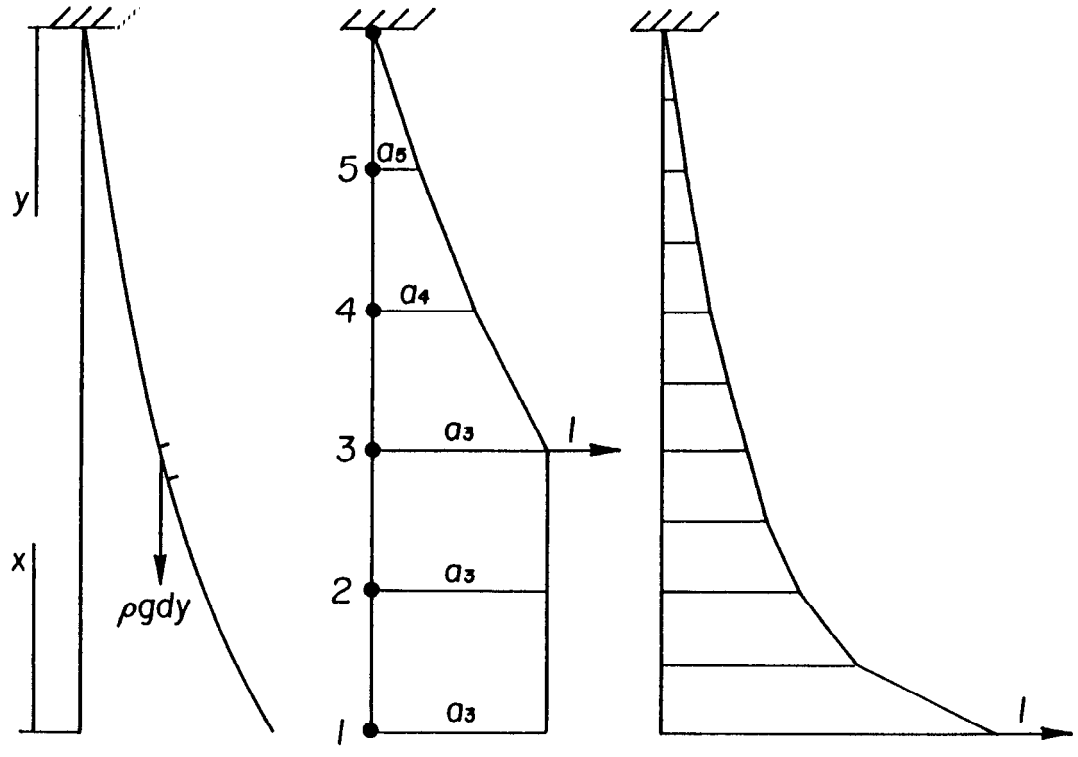
Once more K is positive definite. Otherwise

$$\begin{gathered} 2u^T K u = (u_2 - u_1)^2 + 3(u_3 - u_2)^2 + 5(u_4 - u_3)^2 \\ +7(u_5 - u_4)^2 + 9(u_6 - u_5)^2,\ u_6 = 0 \end{gathered} \tag{3.59}$$

which is the finite difference approximation of $\mathcal{E}$ in eq. (3.54), and K is seen also in this way to be positive definite. Indeed, if u is assumed to be linear between nodes j and $j+1$, then for one such interval,

$$\mathcal{E} = \frac{1}{2} u'^2 \int_{x_j}^{x_{j+1}} x dx = \frac{2j-1}{4} (u_{j+1} - u_j)^2 \tag{3.60}$$

and summation over all intervals produces (3.59).



(a) (b) (c)

Fig. 3.11

Symmetric flexibility matrix F of the hanging string is of the form

$$F = K^{-1} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 & a_5 \\ \times & a_2 & a_3 & a_4 & a_5 \\ \times & \times & a_3 & a_4 & a_5 \\ \times & \times & \times & a_4 & a_5 \\ \times & \times & \times & \times & a_5 \end{bmatrix} \tag{3.61}$$

explained with reference to Fig. 3.11(b). Here

$$F_{ij} = 2\sum_{k=j}^{n} \frac{1}{2k-1}, j \geq i \tag{3.62}$$

and in particular

$$F_{11} = 2\left(\frac{1}{1} + \frac{1}{3} + \frac{1}{5} + \frac{1}{7} + \ldots + \frac{1}{2n-1}\right) \tag{3.63}$$

which keeps increasing very slowly, ultimately like $ln(n)$, as n is increased.

The appearance of a logarithmic term in the flexibility matrix should not surprise us; the fundamental solution to $(xu')' = 0$ also contains a logarithmic term. The first column of F includes the response of the string to a tip point force $f = [1\ 0\ \ldots\ 0]^T$, and the analytic solution of $(xu')' = 0, x \neq 0, u(1) = 0$ is

$$u(x) = c_1 \ln x. \tag{3.64}$$

In the presence of a transverse force the assumption of small displacements becomes unfulfillable at the loose end which is a *singular point* of the hanging string problem. Singularities of this kind are the price of linearization, and they disappear in a nonlinear model that allows for finite displacements and rotations. We defer the discussion of nonlinearity to the last chapter of this book.

Equation (5.64) implies that a *displacement* imposed on the loose end of the string is not transmitted upwards. A numerical solution to $(xu')' = 0, 0 < x < 1, u(0) = 1, u(1) = 0$, however, will be very slow to recognize and simulate the displacement discontinuity at $x = 0$.

The computed tip rotation resulting from the end point loading of Fig. 3.11(c) is given by $(u_1 - u_2)/h = 1/h$, and as $h \to 0$, the tip inclination tends to the horizontal, the reason being that since tension is absent at the lower end point, the string's end must rotate 90^o to tangentially meet the horizontal force.

How does the string respond to a uniformly distributed load $f = 1$? Boundary value problem (3.48) yields in this case

$$u'(x) = -1 + c_1/x, u = -x + c_1 \ln x + c_2 \tag{3.65}$$

where c_1 and c_2 are arbitrary constants of integration. Immovability at $x = 1$ is accounted for with $c_2 = 1$, but the zero slope condition is impossible at $x = 0$. However, the physically plausible condition of finite displacements is achievable with $c_1 = 0$, with which the displacement of the hanging string under $f(x) = 1$ becomes $u(x) = -x + 1$. Boundary condition $u'(0) = 0$ is not fulfilled, but the slope condition can be overruled as it may be discontinuous. In the presence of a skew tip *point* force the slope condition becomes that of tangentiality to the force, but the loose end of the hanging string is free of external forces and tension.

Solution of system (3.57) with $f = 1$ also yields the linear $u = h[n\ n-1\ \ldots\ 1]^T$.

For $f(x) = x$, the differential equation of equilibrium of the hanging brings

$$u'(x) = -\frac{1}{2}x + \frac{c_1}{x},\ u(x) = -\frac{1}{4}x^2 + c_1 \ln x + c_2 \tag{3.66}$$

and $c_1 = 0, c_2 = 1/4$ assure that $u'(0) = 0$ and $u(1) = 0$. Now that $f(0) = 0$, the zero slope boundary condition at the free end of the string can be enforced.

For stiffness matrix K in eq.(3.57) and flexibility matrix F in eq.(3.61) we compute the norms

$$\|K\|_\infty = 4(n-2), \|K^{-1}\|_\infty = S_n,\ S_n = \frac{2}{1} + \frac{4}{3} + \frac{6}{5} + \ldots + \frac{2n}{2n-1} \tag{3.67}$$

where, approximately, $S_n = n + \ln(n)$, so that

$$\kappa_\infty = \|K\|_\infty \|K^{-1}\|_\infty = 4n^2 \tag{3.68}$$

approximately.

A hanging string with *quadratically* varying tension due to a linearly varying density is also of interest. Its equilibrium is described by

$$(x^2 u')' + f = 0 =\ 0 < x < 1,\ u'(0) = u(1) = 0 \tag{3.69}$$

and, using finite differences, we set up for it the stiffness matrix

$$K = K(n \times n) = \frac{h}{4}\begin{bmatrix} 1 & -1 & & & \\ -1 & 10 & -9 & & \\ & -9 & 34 & -25 & \\ & & -25 & 74 & -49 \\ & & & -49 & 130 \end{bmatrix}, h = \frac{1}{n} \tag{3.70}$$

where h renders the right side of the stiffness equation $Ku = f$ to mean force.

Matrix K is factored as

$$K = \frac{h}{4}\begin{bmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & -1 & 1 & \\ & & & -1 & 1 \end{bmatrix}\begin{bmatrix} 1 & & & & \\ & 9 & & & \\ & & 25 & & \\ & & & 49 & \\ & & & & 81 \end{bmatrix}\begin{bmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & 1 & -1 & \\ & & & 1 & -1 \\ & & & & 1 \end{bmatrix} \tag{3.71}$$

and we verify thereby that the matrix is positive definite. Flexibility matrix $F = K^{-1}$,

$$K^{-1} = \frac{4}{h}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ & 1 & 1 & 1 & 1 \\ & & 1 & 1 & 1 \\ & & & 1 & 1 \\ & & & & 1 \end{bmatrix}\begin{bmatrix} 1 & & & & \\ & 1/9 & & & \\ & & 1/25 & & \\ & & & 1/49 & \\ & & & & 1/81 \end{bmatrix}\begin{bmatrix} 1 & & & & \\ 1 & 1 & & & \\ 1 & 1 & 1 & & \\ 1 & 1 & 1 & 1 & \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.72}$$

is here of the same form as that in eq. (3.61) and we readily establish that

$$F_{ii} = 4n\sum_{k=i}^{n}\frac{1}{(2k-1)^2},$$

$$F_{nn} = 4n\frac{1}{(2n-1)^2}, \quad \text{and} \quad F_{11} = 4n(1 + \frac{1}{3^2} + \frac{1}{5^2} + \ldots + \frac{1}{(2n-1)^2}). \tag{3.73}$$

The sum of the series in the expression for F_{11} never exceeds $\pi^2/8$, and for large $n, F_{nn} = 1/n$, and $F_{11} = n\pi^2/2$, approximately.

The large $F_{11} = n\pi^2/2$ is due to a stronger singularity at the free end of the hanging string with $p(x) = x^2$ than that of the hanging string with $p(x) = x$, the solution of $(x^2u')' = 0$ being

$$u'(x) = c_1/x^2, \; u(x) = -c_1/x + c_2. \tag{3.74}$$

A hanging string with quadratically varying tension cannot carry even a uniform load. If $(x^2u')' = -1, \; 0 < x < 1$, then

$$u'(x) = -1/x + c_1/x^2, \; u(x) = -\ln x - c_1/x + c_2 \tag{3.75}$$

and no choice of c_1 and c_2 renders $u(0)$ finite. In the case of $f = x$,

$$u'(x) = -\frac{1}{2} + \frac{c_1}{x^2},\ u(x) = -\frac{1}{2}x + \frac{c_1}{x} + c_2 \tag{3.76}$$

and the displacement is finite with $c_1 = 0$. But $f(x) = x^2$ allows the satisfaction of both $u(1) = 0$ and $u'(0) = 0$.

Stiffness matrix K for the hanging string with a quadratically varying tension, given in eq. (3.70), and its inverse have the norms

$$\|K\|_\infty = 4n,\ \|K^{-1}\|_\infty = 2nS_n, \kappa_\infty = 4n^2 \ln(n) \tag{3.77}$$

since

$$S_n = \frac{2}{1} + \frac{4}{9} + \frac{6}{25} + \ldots + \frac{2n}{(2n-1)^2} \tag{3.78}$$

is $\ln(n)/2$, approximately.

Exercises

3.6.1. A string of unit length hangs with a weight W_1 attached to its lower free end, and a weight W_2 attached at midpoint. Ignoring self weight, the tension in the string is given by

$$p(x) = W_1 \quad 0 \le x < \frac{1}{2}\ ,\ p(x) = W_1 + W_2 \quad \frac{1}{2} \le x \le 1.$$

Write the differential equation of equilibrium for the string and discretize it by finite differences. Compare the approximate solution with the exact.

3.6.2. Is the inverse of matrix K in eq. (3.35) positive? Compute $\kappa_\infty = \|K\|_\infty \|K^{-1}\|_\infty$ for this matrix.

3.6.3. For the finite difference model of $(xu')' = 0$, $0 < x < 1$, $u(0) = 1/2$, $u(1) = 0$, study the approximate deflection of the string at $x = 1/2$ as a function of the number n of nodes.

11. A string of density ρ per unit length is attached to the axis of a turntable that revolves with an angular velocity ω. Assume the string is in radial position and write the tension it is under.

3.7 Fourth-order problem–bent beam

Physical perspectives underscore our present discussion, and we shall explore the linear algebraic properties of the discrete model for a fourth-order problem on the physically comprehensible *beam-bending problem*.

Consider a long metal or wood rod of rectangular cross-section, resting on two sharp supports at each end and laterally loaded by a distributed force $f(x)$, as shown in Fig. 3.12. As a result of the action of load $f(x)$ the rod bends and assumes configuration $u(x)$. Deformations are here entirely elastic but the beam as a whole is assumed to be axially inextensional, and hence the right rolling support. In the linear analysis displacements are assumed to be small compared to the thickness of the beam. Forces are transmitted through the beam to the supports not by axial tension, which is entirely absent here, but by lateral cross-sectional *sheer stresses* and *bending moments*.

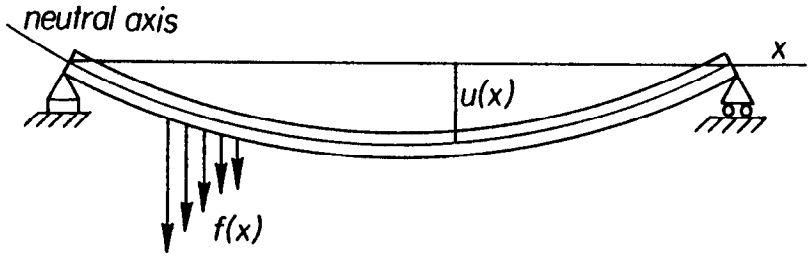


Fig. 3.12

As the beam flexes, longitudinal material fibers above a *neutral axis* shorten, while fibers below it lengthen as in Fig. 3.13(a). The neutral axis runs parallel to the long sides of the beam and it is reasonably assumed (the Kirchhoff assumption) that cross sections remain plane and normal to the axis during bending. We also assume that elastic deformations are occurring longitudinally only.

At point x the radius of curvature of the beam is given by

$$r = \frac{(1 + u'^2)^{3/2}}{u''} = \frac{1}{u''} \quad \text{if } |u'| << 1 \tag{3.79}$$

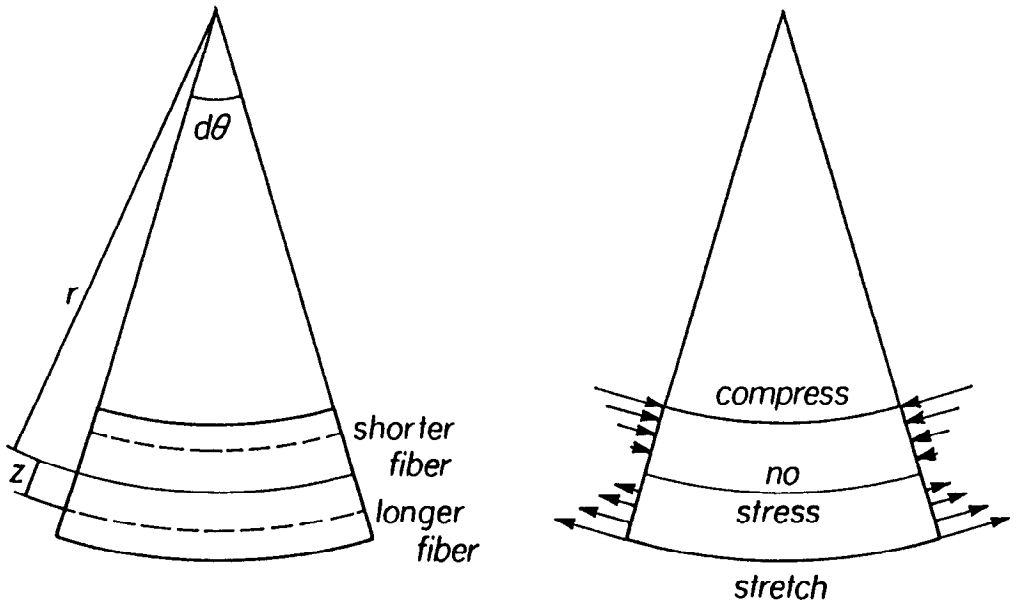
and the relative elongation of a fiber located at distance z from the neutral axis is

$$\epsilon = \frac{(r + z)d\theta - rd\theta}{rd\theta} = zu''. \tag{3.80}$$

It is further assumed that the beam material is *linearly elastic* so that the elastic restoring stress σ (force per unit area) is proportional to ϵ, or

$$\sigma = E\epsilon = Ezu'' \tag{3.81}$$

where E is the *elastic modulus* of the material. Figure 3.13(b) shows the normal stress distribution on typical cross sections. In the absence of axial forces and with a rectangular cross section, the neutral axis passes symmetrically midway between the long edges.



(a) Fig. 3.13 (b)

A resultant axial force is absent here, but due to the variable σ, there appears at any typical cross section a bending moment M equal to

$$M = b\int_{-t/2}^{t/2} \sigma z dz = Eu'' b\int_{-t/2}^{t/2} z^2 dz = E\frac{bt^3}{12}u'' \tag{3.82}$$

where b is the width of the beam and t its thickness. In short

$$M = EIu'' \tag{3.83}$$

where $I = bt^3/12$ is the beam's cross-sectional *moment of inertia*.

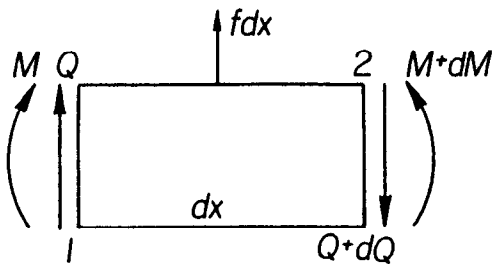


Fig. 3.14

We assume the lateral force $f(x)$ to be such that it never causes a corner or a discontinuous slope u'. The beam can carry a point *force* and even a point *moment*.

Apart from the bending moment a reactive shearing force Q arises in the beam as a result of the applied load as in Fig. 3.14. Vertical zero force sum yields the first equilibrium equation

$$Q + f dx - Q - dQ = 0 \quad \text{or} \quad Q' = f \tag{3.84}$$

for the beam differential element dx, while zero moment sum about point 2 yields the second

$$M + Q dx + f dx \; dx/2 - M - dM = 0 \quad \text{or} \quad Q = M' \tag{3.85}$$

if $(dx)^2$ is neglected. The combination of eqs. (3.84) and (3.85) result in

$$M'' = f(x) \tag{3.86}$$

becoming with eq. (3.83)

$$EIu'''' = f(x) \tag{3.87}$$

which is a linear fourth-order equation with constant coefficients.

Common homogeneous displacement and force boundary, or edge, conditions for the beam are shown in Fig. 3.15.

All energy stored in the deformed beam is elastic, and to compute it we consider a fiber element of length dx and thickness dz additionally extended from 0 to ϵ. The differential increase in elastic energy of the element is

$$d\mathcal{E} = b dx dz \int_0^\epsilon \sigma d\epsilon \tag{3.88}$$

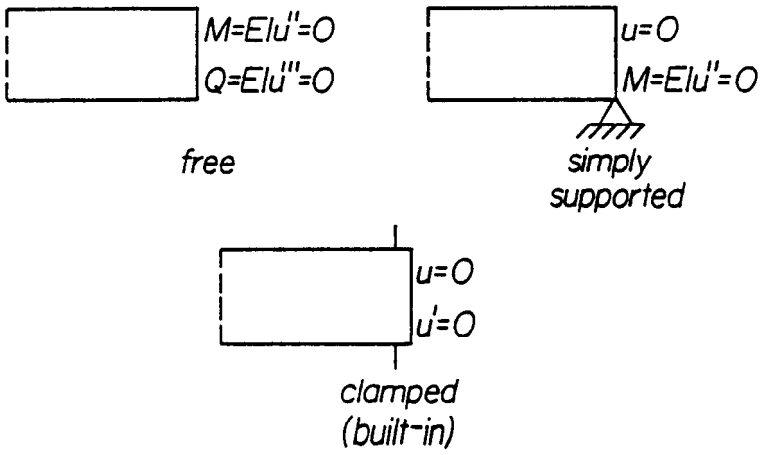


Fig. 3.15

which with $\sigma = E\epsilon$ becomes

$$d\mathcal{E} = \frac{b}{2} E\epsilon^2 dxdz \tag{3.89}$$

and for the entire beam of length L and thickness t

$$\mathcal{E} = \frac{b}{2} E \int_0^L \int_{-\frac{1}{2}t}^{\frac{1}{2}t} \epsilon^2 dxdz. \tag{3.90}$$

But $\epsilon = zu''$, and

$$\mathcal{E} = \frac{1}{2} EI \int_0^L u''^2 dx \tag{3.91}$$

where $I = bt^3/12$.

A metal piano string designed to produce loud tones has the mixed properties of beam and string. Acoustically it is preferable that it have the nature of an ideal string, and by dint of the great tension administered to it, it indeed behaves more like a string than a beam. But even forcing the string down laterally with a sharp object does not cause it to bend sharply as would happen to a linen thread.

A finite difference formula for u'''' requires u at five nodes and we write at node j the approximations

$$u_j'' = \frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}), u_j'''' = \frac{1}{h^2}(u_{j-1}'' - 2u_j'' + u_{j+1}'') \tag{3.92}$$

so that for, say, point 3

$$u_3'''' = \frac{1}{h^4}(u_1 - 4u_2 + 6u_3 - 4u_4 + u_5). \tag{3.93}$$

We shall now use this formula to approximate the fourth-order, two-point boundary value problem

$$\begin{aligned} u'''' &= f \quad 0 < x < 1 \\ u''(0) &= u'''(0) = 0 \\ u(1) &= u'(1) = 0 \end{aligned} \tag{3.94}$$

that describes the deformation of a beam free at $x = 0$ and clamped at $x = 1$. We shall also consider boundary conditions $u(1) = u''(1) = 0$ for a simple support at $x = 1$.

With reference to the grid in Fig. 3.5 we write:

$$\begin{aligned} &\text{at node 1} \quad \frac{1}{h^4}(u_{-1} - 4u_0 + 6u_1 - 4u_2 + u_3) = f_1, \\ &\text{at node 2} \quad \frac{1}{h^4}(u_0 - 4u_1 + 6u_2 - 4u_3 + u_4) = f_2. \end{aligned} \tag{3.95}$$

Boundary conditions $u''(0) = u'''(0) = 0$ at node 1 are approximated by

$$u_j'' = \frac{1}{h^2}(u_{j-1} - 2u_j + u_{j+1}) \quad j = 0, 1, 2 \;\; u_1'' = 0 \tag{3.96}$$

and

$$u_1''' = \frac{1}{2h}(u_2'' - u_0'') = 0 \tag{3.97}$$

resulting in

$$u_0 = 2u_1 - u_2 \;\; \text{and} \;\; u_{-1} = 4u_1 - 4u_2 + u_3 \tag{3.98}$$

which permit the elimination of u_0 and u_{-1} from the two equations (3.95). Now:

$$\begin{aligned} &\text{at node 1} \quad \frac{1}{h^4}(u_1 - 2u_2 + u_3) = \frac{1}{2}f_1, \\ &\text{at node 2} \quad \frac{1}{h^4}(-2u_1 + 5u_2 - 4u_3 + u_4) = f_2, \end{aligned} \tag{3.99}$$

and for the entire discretization

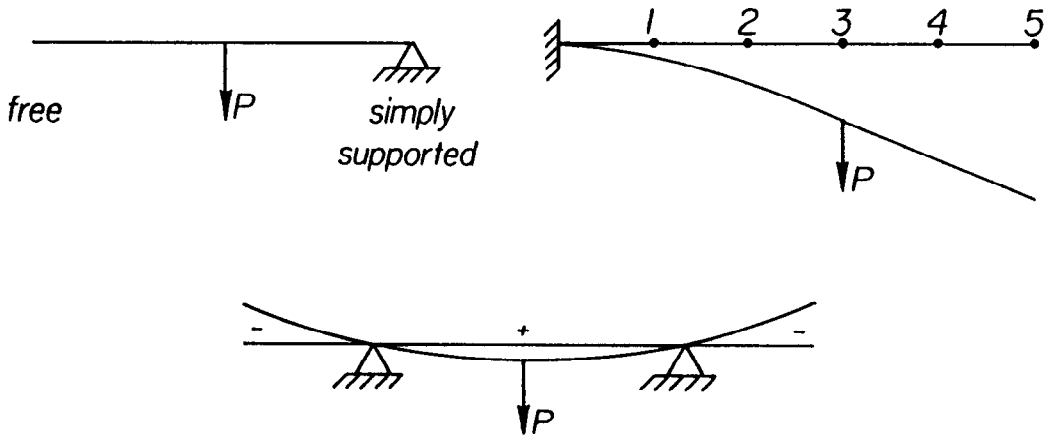
$$\frac{1}{h^3}\begin{bmatrix} 1 & -2 & 1 & & & \\ -2 & 5 & -4 & 1 & & \\ 1 & -4 & 6 & -4 & 1 & \\ & 1 & -4 & 6 & -4 & 1 \\ & & 1 & -4 & 6 & -4 \\ & & & 1 & -4 & \beta \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} = h \begin{bmatrix} \frac{1}{2}f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix}, Ku = f \tag{3.100}$$

where $\beta = 5$ if the right end of the beam is simply supported, $u(1) = u''(1) = 0$; and $\beta = 7$ if the right end is clamped, $u(1) = u'(1) = 0$.

Triangular factorization produces for K in eq. (3.100)

$$h^3K = LL^T \ , L = \begin{bmatrix} 1 & & & & & \\ -2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & 1 & -2 & 1 & \\ & & & 1 & -2 & \alpha \end{bmatrix} \tag{3.101}$$

in which $\alpha = \sqrt{2}$ if $u(1) = u'(1) = 0$, and K is then positive definite. If $u(1) = u''(1) = 0$, if the beam is simply supported at the right end point (recall that it is free at the left end), then $\alpha = 0$ and K is singular, and only positive semidefinite. Obviously no elastic solution exists for these latter boundary conditions as is seen in Fig. 3.16(a).



(a) (c) (b)

Fig. 3.16

The flexibility matrix for the beam is computed from eq.(3.101) for $\alpha = \sqrt{2}$, in the form

$$F = K^{-1} = h^3L^{-T}L^{-1}, L^{-1} = \begin{bmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ 3 & 2 & 1 & & & \\ 4 & 3 & 2 & 1 & & \\ 5 & 4 & 3 & 2 & 1 & \\ 6/\alpha & 5/\alpha & 4/\alpha & 3/\alpha & 2/\alpha & 1/\alpha \end{bmatrix} \tag{3.102}$$

and F is once more *dense* and *positive*. A point force at any node causes all points of the beam to displace in the direction of the force as in Fig. 3.16(b).

But F need not always be positive for a beam problem. Figure 3.16(c) shows a beam on *internal* supports. Rotation is transmitted over a simple support and a positive deflection

between the supports raises the overhanging sections in the negative direction. Positive definiteness is a deeper, and more general property of the flexibility matrix than the positiveness of all entries, which is not always there.

That K in eq.(3.100) is positive definite when $\beta = 7$ may also be proved by establishing that

$$\begin{aligned} h^3 u^T K u = &(u_1 - 2u_2 + u_3)^2 + (u_2 - 2u_3 + u_4)^2 + \ldots + (u_5 - 2u_6 + u_7)^2 \\ &+ \frac{1}{2}(u_6 - 2u_7 + u_8)^2 \end{aligned} \tag{3.103}$$

in which $u_7 = 0$ and $u_8 - u_6 = 0$ to satisfy the boundary conditions $u(1) = u'(1) = 0$. The last term in eq.(3.103) is then $2u_6^2$, and the one before it $(u_5 - 2u_6)^2$. It follows that $u^T K u = 0$ only if $u_6 = 0, u_5 = 0, u_4 = 0 \ldots u_1 = 0$, otherwise $u^T K u > 0$, and K is positive definite.

If the beam deflection is assumed parabolic over the interval $2h$ between, say nodes 1 and 3, then $u'' = (u_1 - 2u_2 + u_3)/h^2$ over this interval and the elastic energy of the segment is according to eq. (3.91)

$$\mathcal{E} = \frac{1}{2h^3}(u_1 - 2u_2 + u_3)^2 \tag{3.104}$$

if $EI = 1$. Or with $u = [u_1 \; u_2 \; u_3]^T$

$$\mathcal{E} = \frac{1}{2} u^T k u, \;\; k = \frac{1}{h^3} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}. \tag{3.105}$$

We interpret eqs.(3.103), (3.104) and (3.105) as implying that the elastic energy of the beam consists of the sum of overlapping parabolic segments similar to what is shown in Fig. 3.6. Only half the energy of the last segment, that extends over the right-hand clamping support, is added, and $u_7 = 0$, and $u_6 = u_8$ are imposed on it. Also, because node 1 is at a free end where $M = Q = 0$, it happens that $u_0 - 2u_1 + u_2 = 0$.

To compute the infinity norm of the beam flexibility matrix F we take advantage of its positiveness, solve $Ku = e, e = [1 \; 1 \; \ldots \; 1]^T$, using the LL^T factorization of eq. (3.102), and obtain

$$\|K^{-1}\|_\infty = u_1 = \frac{1}{4} h^3 (2 \sum_{j=1}^{n} (j^2 + j^3) - n^2(n+1)) \tag{3.106}$$

which by the known summation formulas for j^2 and j^3, and after ignoring n^2 and n^3 relative to n^4, leaves us with

$$\|K\|_\infty = 16n^3, \ \|K^{-1}\|_\infty = \frac{1}{8}n, \ \kappa_\infty = 2n^4. \tag{3.107}$$

The beam stiffness matrix has an infinity condition number κ_∞ that grows proportionally to n^4, while that of the string grows proportionally to n^2 only. The beam stiffness matrix becomes ill-conditioned much faster than that of the string, and this is one of the most important computational distinctions between the two problems.

On the other hand, solution of $u'''' = f$ involves four repeated integrations of f, and as a result deflection u of the beam is of a higher degree of continuity than that of the string, allowing for a coarser mesh with fewer unknowns. If f is a concentrated point force, or a delta function, then the two repeated integrations needed to produce $u = u(x)$ for the string from $-u'' = f$ assure the deflection to be continuous, but not more. The four repeated integrations of the beam equation produce a deflection curve that is, in this case, not only continuous in itself but even continuous in the second derivative.

Exercises

3.7.1. Write the stiffness matrix for a beam with an internal hinge as in Fig. 3.17.

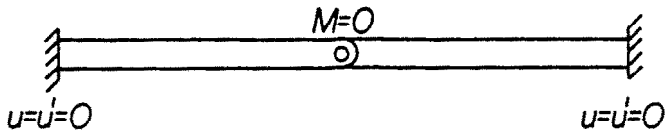


Fig. 3.17

3.8 Two- and three-dimensional problems

Both the string and beam problems allowed us to gain invaluable insights into the physical background of the basic linear algebraic properties of their discrete equations of equilibrium. Because these problems are only one-dimensional we were able to write out explicitly the stiffness and flexibility matrices for them . We observed the band form of the stiffness matrix,

its positive definiteness, and the inevitable increase in size and bandwidth in response to a quest for greater accuracy, and we considered the question of why the flexibility matrix is dense and often with entirely positive entries.

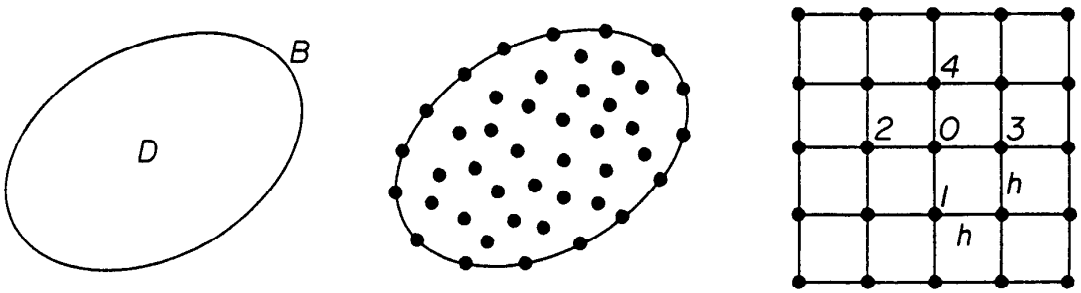
Two- and three-dimensional field problems are cumbersome and we can consider them here in broad generalities only.

Corresponding to the taut string deflection problem $u'' + f(x) = 0$, with appropriate boundary conditions, are the *partial* differential equations

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x,y) = 0 \text{ and } \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 y}{\partial z^2} + f(x,y,z) = 0 \tag{3.108}$$

in plane and space, respectively. They are the most common equations of equilibrium in linear mathematical physics and their discretization should amply illustrate to us the basic computational issues raised by higher dimensions.

Take first $u_{xx} + u_{yy} + f = 0$. Among other things it describes the deflection of a thin taut *membrane* by a surface-distributed load f, and we want to think of the boundary value problem in terms of this physical allusion. The membrane extends over domain D with boundary B on which some boundary conditions, say $u = 0$, prevail. Discretization of the membrane boundary value problem of Fig. 3.18(a) consists of replacing the partial differential equation of equilibrium at each point of a set of nodes in D and on B, as in Fig. 3.18(b), by a finite difference approximation involving assumed displacements u at the point and some nodes around it. Precisely how to do it, particularly how to accommodate boundary B and the u conditions on it, is beyond our present resources. Finite element discretization techniques handle these questions with great methodological astuteness.



(a) (b) (c)

Fig. 3.18

It is enough for our purpose that we consider the square domain in Fig. 3.18(c) with nodes regularly placed on grid lines parallel to the sides. At typical interior node 0 we write

$$u_{xx} + u_{yy} = \frac{1}{h^2}(u_1 + u_2 + u_3 + u_4 - 4u_0) \tag{3.109}$$

and accept that a higher-order finite difference approximation to $u_{xx} + u_{yy}$ would involve more nodes around grid point 0.

Discretization of the entire membrane boundary value problem is accomplished by writing the finite difference approximation at all interior points and imposing the prescribed boundary conditions.

Clearly, the stiffness matrix for the membrane is sparse because equilibrium is pointwise, and each equation includes only few neighboring nodal values. We also foresee the flexibility matrix $F = K^{-1}$ as being completely dense. But the *sparseness pattern* of the matrix greatly depends on the way the nodes are numbered, a task more involved and infinitely richer in possibilities here than in one dimension. The rest of this chapter is devoted to node numbering strategies designed to achieve certain declared storage and computational objectives for stiffness matrix K.

Matters are more involved in space where there are layers upon layers of grid planes. Matrices are still sparse and basically of band form, but everything is on a gigantic scale.

The fourth-order beam problem $u'''' = f$ becomes in the plane the biharmonic

$$\frac{\partial^4 u}{\partial x^4} + 2\frac{\partial^4 u}{\partial x^2 \partial y^2} + \frac{\partial^4 u}{\partial y^4} = f(x, y) \tag{3.110}$$

equation of equilibrium for the *thin elastic plate*. Finite difference approximation of the plate equation does not differ in principle from that of the membrane.

In case of polar and spherical symmetries, equations (3.108) reduce to

$$(ru')' + rf(r) = 0 \text{ and } (r^2u')' + r^2 f(r) = 0,\ 0 < r < 1 \tag{3.111}$$

in two and three dimensions, respectively, reminding us of the equations of equilibrium of the hanging string with linear and quadratic tensions.

The method of finite elements, universally used to discretize boundary value problems of the membrane and plate kind, as well as the more general elastic problem, produces fittingly

and without exception symmetric positive (semi)definite stiffness matrices. We shall consider finite elements in the last chapter of the book, but meanwhile we assume K to be symmetric and positive definite, always admitting an LL^T factorization.

Exercises

3.8.1. Use the approximation

$$au_{xx} + bu_{yy} = \frac{1}{h^2}(a(u_2 - 2u_0 + u_3) + b(u_1 - 2u_0 + u_4))$$

to write the stiffness equation for the square membrane problem

$$u_{xx} + u_{yy} + f(x, y) = 0$$

in the unit square of Fig. 3.19

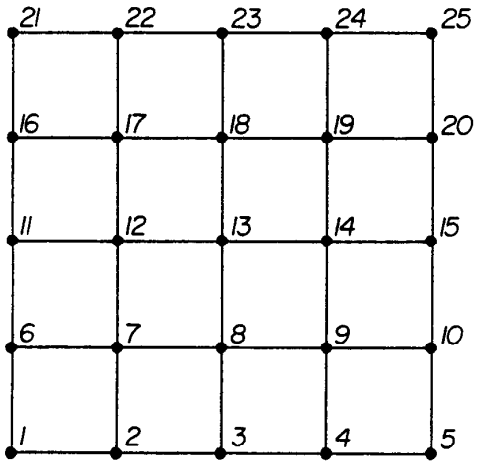


Fig. 3.19

with $u = 0$ on all four edges.

Verify that K is positive definite and symmetric, and that $F = K^{-1}$ is dense and positive. Consider the mechanical model with short ties or links. Compute κ_∞ for this $m \times m$ problem, and estimate it as a function of $h = 1/m$. Explain why the response of the membrane is more localized than that of the string under constant tension.

Let the load be symmetric so that $u_7 = u_9 = u_{17} = u_{19}, u_8 = u_{12} = u_{14} = u_{18}$.

Is LL^T the discrete counterpart to $(\frac{\partial}{\partial x} + \frac{\partial}{\partial y})(\frac{\partial}{\partial x} + \frac{\partial}{\partial y})$?

3.8.2. Same as 3.8.1. but for

$$u_{xx} + 2u_{yy} + f(x, y) = 0$$

3.8.3. Same as 3.8.1. but for

$$u_{xx} + 3u_{yy} + 1 = 0$$

with shown boundary conditions

$$\begin{array}{ccc} & \frac{\partial u}{\partial y} + u = 0 & \\ u = 0 & \square & u = 0 \\ & \frac{\partial u}{\partial y} - u = 0 & \end{array}$$

Discuss symmetry issues for K.

One need not get overly involved with difficulties and uncertainties of finite difference approximations. Finite elements, to be briefly discussed in the last chapter, produce stiffness matrices theoretically guaranteed to be symmetric and positive definite.

3.9 Sparseness patterns–band and envelope

Discrete equations of equilibrium have sparse matrices. The tridiagonal string stiffness matrix includes only $3n - 2$ nonzeroes for a total of n^2 entries.

The relationship between the distribution of nonzero entries in stiffness matrix

$$K = \begin{bmatrix} \times & \times & & & & \\ \times & \times & \times & & & \\ & \times & \times & \times & & \\ & & \times & \times & \times & \\ & & & \times & \times & \times \\ & & & & \times & \times \end{bmatrix} \tag{3.112}$$

of the string and the drawing made to mark and number (label) the nodes of the finite difference grid suggests that we call the latter the *graph* of the matrix. The graph consists of n nodes (vertices), of which nodes i and j are said to be *connected* if $K_{ij} = K_{ji} \neq 0$. We are dealing exclusively with symmetric positive definite matrices for which $K_{ii} \neq 0$ and every node is self-connected.

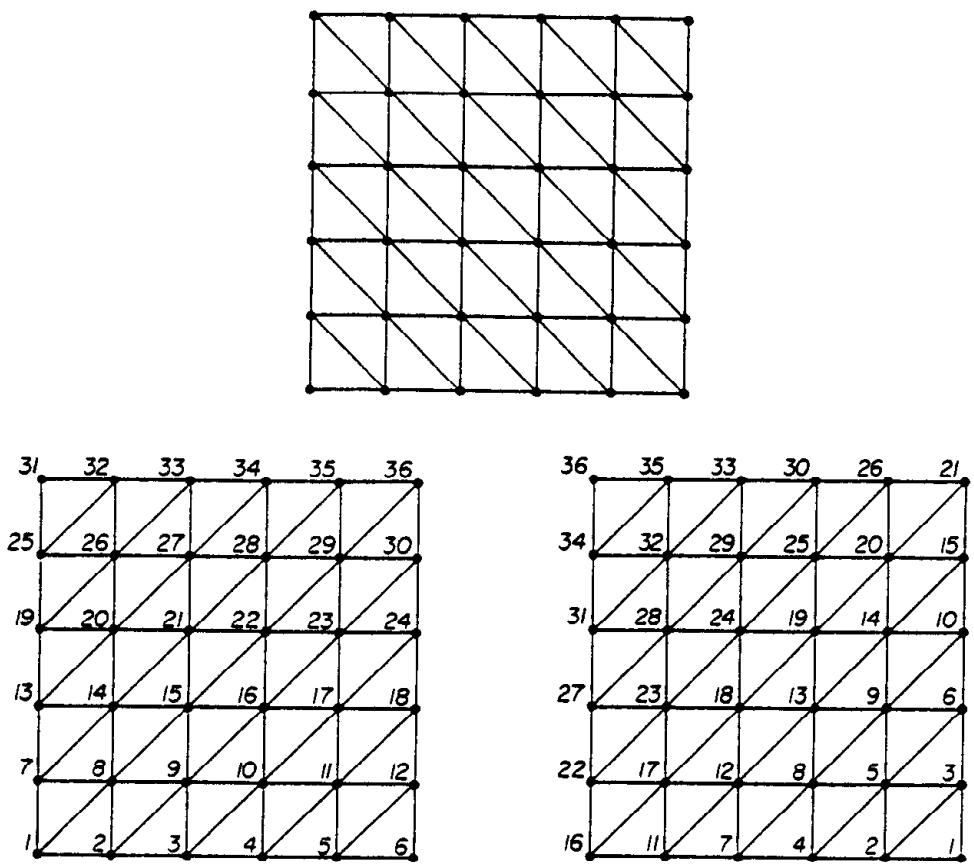
Given a matrix we may draw its graph, but with finite differences, and as we shall see later with finite elements, the graph precedes the matrix and very often has a definite physical existence. Two points on the string are connected if they are end points of a tie in a linkage model.

A graph has only n points and provides concise means by which to visualize the distribution of the nonzero entries in the $n \times n$ matrix. For the one-dimensional string problem it does not occur to us to number the nodes other than consecutively, hence the tridiagonal form of K. A decisive property of the graph of K is that *the interchange of two node labels results in the symmetric permutation of the corresponding rows and columns of symmetric matrix* K. Any other numbering creates an equivalent linear system and is therefore permissible, but it will destroy the band form and disperses the nonzero entries, and this we do not want. We cherish the band form, for it allows an economic organization of the solution of the linear system.

Figure 3.20(a) shows a two-dimensional grid of nodes with their ties or connectors. A node is marked by a full small circle • and there is one unknown at each of the 36 nodes. Nodes are not yet labeled but we know that there are 36 equations of equilibrium for the grid of Fig. 3.20, one per node, each equation including the unknowns at connected nodes only. We may picture the grid as being real with the node connections being actual ties. Such a two- or three- dimensional rod structure is called by engineers a *truss*. One thinks of a truss as a building frame.

Evidently size is a severe computational challenge for discrete systems of equilibrium equations of two- and three-dimensional problems. A grid with 25 points *per side* produces a 625×625 stiffness matrix in two dimensions and a 15625×15625 stiffness matrix in three dimensions. If the grid truly represents a truss or building skeleton, then the movement of each node is determined by three displacements and the number of unknowns triples. And 25 points per side are realistically very few.

Holding and solving full systems of such magnitude is unworkable even for the fastest and largest of computers. Simulation and solution of realistic discrete two and three dimensional models is a pragmatic computational proposition only if full advantage is taken of



(a)

(b) (c)

Fig. 3.20

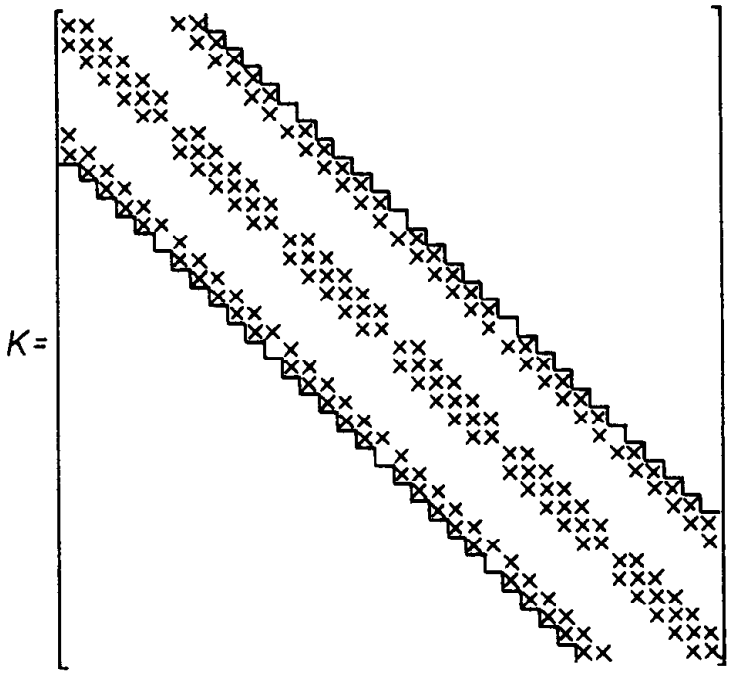
the *sparseness* of K. Computation of the usually dense flexibility matrix $F = K^{-1}$ is out of the question for such large systems, but we have already noted that forward elimination produces *sparse* triangular matrices. Central to the argument of this section is

Theorem 3.1. *If K is a band matrix factored as $K = LDL^T$, then*

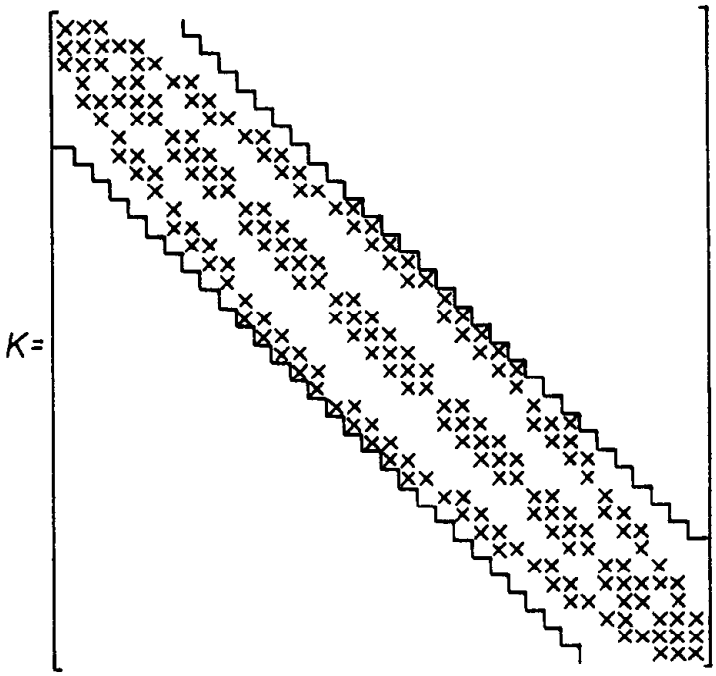
$$band\ (L + L^T) = band\ (K). \tag{3.113}$$

The proof to this theorem rests on the simple observation that forward elimination and back substitution in a band system is entirely confined to the band. Zero entries outside the band need neither be stored nor arithmetically handled.

A primary node labeling objective becomes manifest: *to create K with the smallest band*. Two good band numberings are shown in Fig. 3.20(b) and 3.20(c) with corresponding stiffness matrices in eqs. (3.114) and (3.115)



(3.114)



(3.115)

Both are considered band matrices, with K having a half bandwidth equal to 7 in eq. (3.114), and 6 in eq. (3.115). This is the best we can do, and generally the half bandwidth of K for an $m \times m$ grid is m . Three-dimensional problems are too large for detailed description,

but we envision that an $m \times m \times m$ mesh with stacked planes numbered consecutively produces a matrix of size $m^3 \times m^3$ with half bandwidth equal to m^2.

The second important computational consideration with sparse matrices is *algorithmic simplicity* or low overhead. Gauss solution and the triangular factorization of K is as simple for band systems as for a dense system, and hence the appeal of band storage. The band itself is sparse but many of its zero entries turn nonzero during forward elimination; matrix L suffers *fill*. Equations (3.116) and (3.117) show L^T in $K = LL^T$ for K in eqs. (3.114) and (3.115), respectively with an $\times$ to mark an original nonzero entry in K, and a $\bullet$ to mark a nonzero created during factorization.

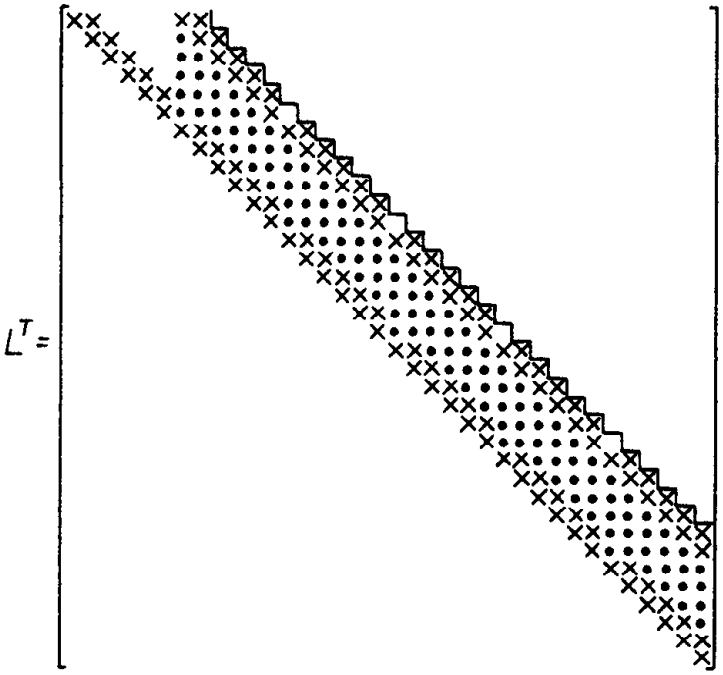
In the simplest band version of Gauss elimination, the pivots are taken consecutively on the diagonal and are used to eliminate the current unknown from all equations below it *inside* the band. No attempt is made to distinguish between permanent zeros and nonzero entries inside the band. This is the chief algorithmic simplification of the solution procedure. But from purely arithmetical considerations it is not entirely *efficient*. Some band entries in eq. (3.116), and more in eq. (3.117) are permanently zero and it is arithmetically wasteful to operate with them. A more sophisticated solution procedure would perform a symbolic factorization first and then avoid all null arithmetical operations.

But here ultimate arithmetical efficiency is being paid for with a more complicated, costlier, overhead-burdened algorithm.

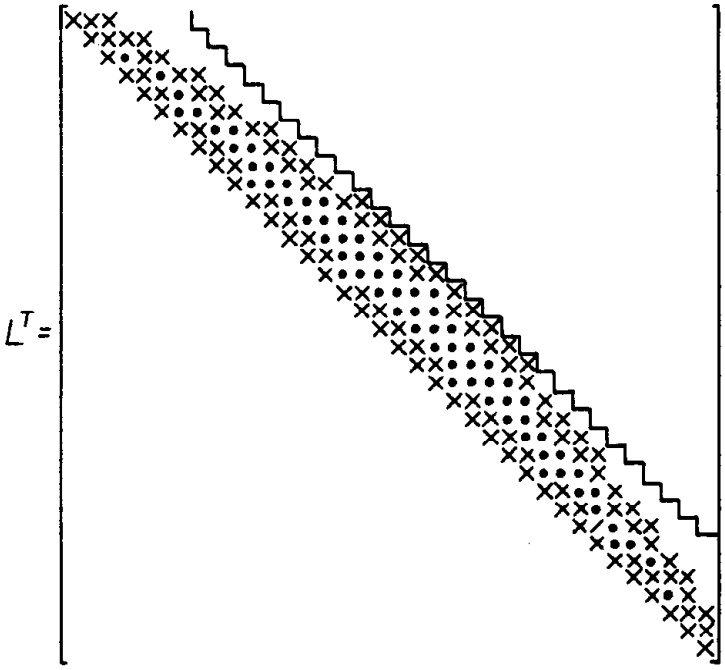
For a rectangular grid it is rather obvious how to number the nodes for a narrow band. Bandwise, the labeling rule of Fig. 3.21(a) is unmistakably good whereas that of Fig. 3.21(b) is bad.

For grids with extensions and cutouts one expects to do better than a bandwidth for a covering rectangle, yet no renumbering algorithm exists to find the very minimal bandwidth in reasonable (polynomial) time, and to try all possibilities takes forever. Because band reduction through node renumbering is of such great computational interest, *heuristic* algorithms have been devised to search for improved labeling to reduce bandwidth.

Labeling uncertainties and grid irregularities combine to erode the computational efficiency of the band algorithm. The next step in algorithm sophistication is a *variable* band-

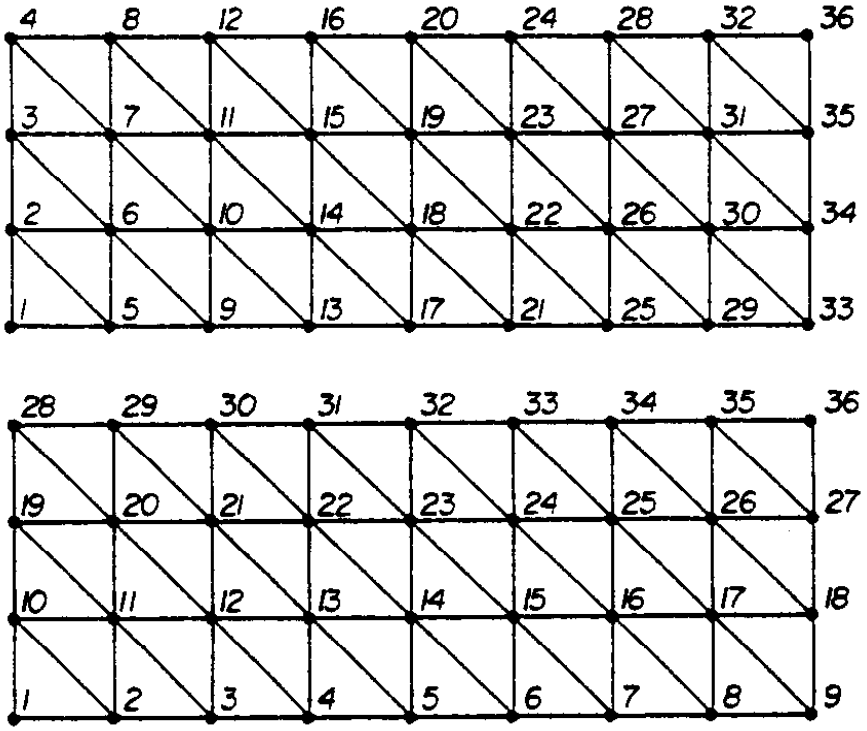


(3.116)



(3.117)

width or *envelope* (also *skyline*) storage. Figures 3.22(a),(b) and (c) show an L-shaped grid and a circular grid with two different node numberings. The stiffness matrices produced by the grids are, correspondingly, written in eqs. (3.118),(3.119), and (3.120), with the envelope



top (a) Fig. 3.21 bottom (b)

heavily outlined.

Symmetric positive definite matrices are such that no zero pivot ever appears to require row and column permutations. Gauss solution of the linear system works with entries within the envelope only. We state it formally in

Theorem 3.2. *If K is a symmetric and positive definite matrix factored as $K = LL^T$, then*

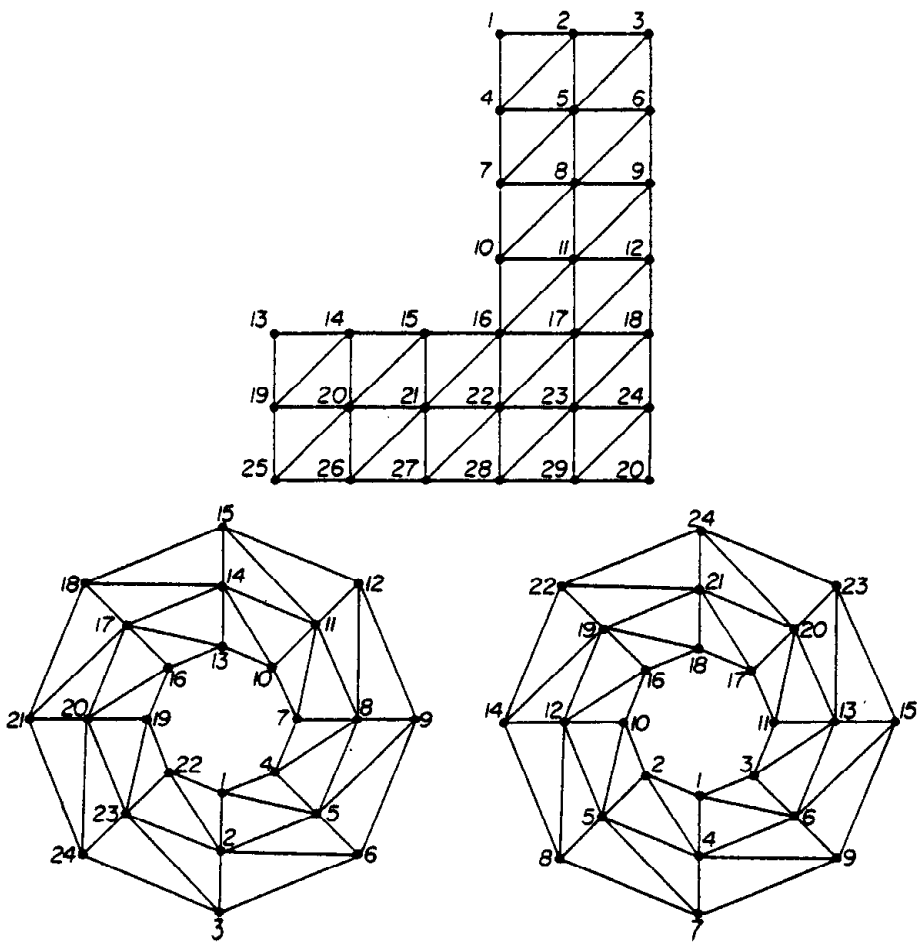
$$envelope\ (L + L^T) = envelope\ (K). \tag{3.121}$$

Envelope algorithms are more efficient than band algorithms but are not considerably more complicated and are favored in commercial finite difference and finite element codes.

The upper-triangular L^T in the symbolic factorizations of K in eqs. (3.118),(3.119), and (3.120) is given in eqs. (3.122),(3.123) and (3.124), respectively.

3.10 Sparse algorithms

These take ultimate advantage of the matrix sparseness. Only nonzero entries in the LL^T factorization of K are stored and arithmetically handled. All other zero entries are ignored.



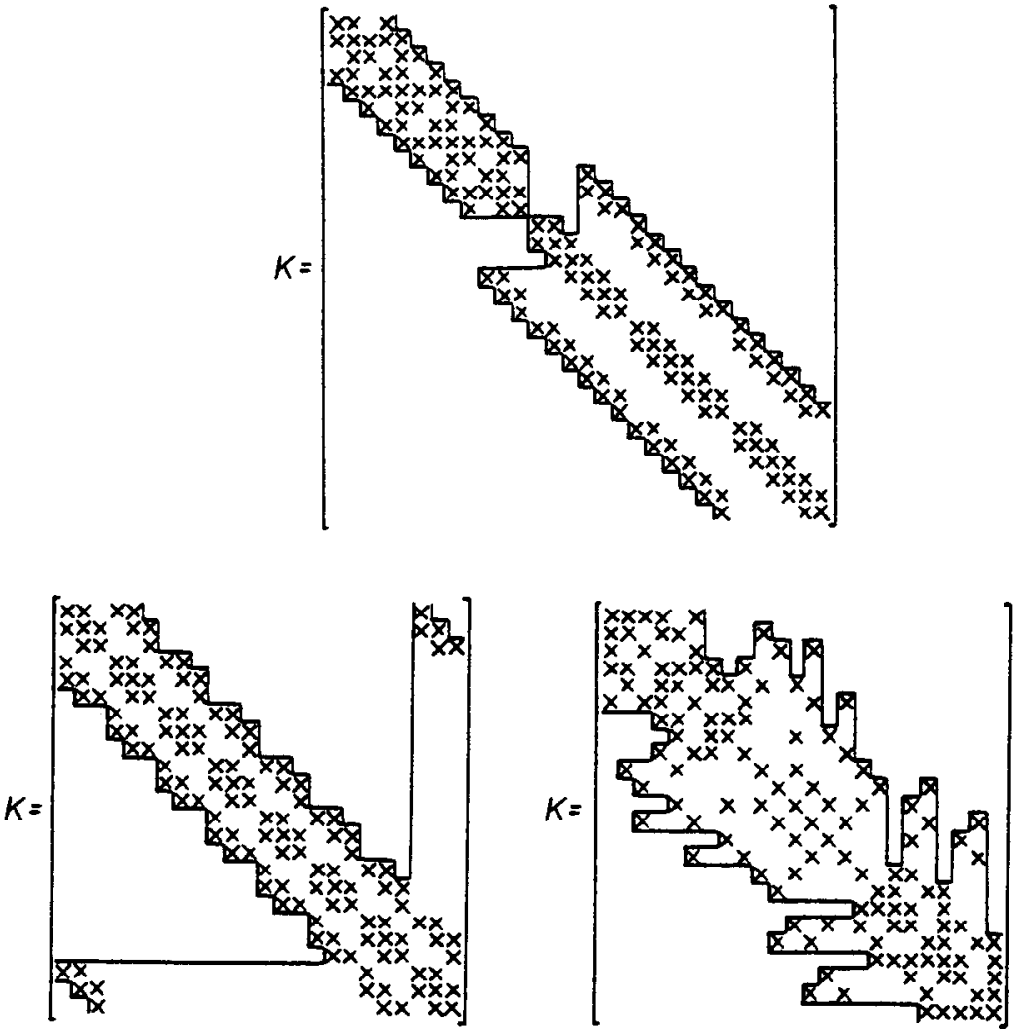
(a)

(b) (c)

Fig. 3.22

Before the arithmetic starts matrix K is *symbolically* factored (assuming nodal numbering is decided upon) to determine the nonzero entries of L, and only they are stored. This requires some sophisticated bookkeeping to relate the densely stored entries of K to their location in the two-dimensional table. Computational overhead both in storage and arithmetic is incurred thereby and should be included in the overall storage and cost assessment of the algorithm.

The avowed labeling objective of sparse algorithms is to *reduce fill*; to have the least number of nonzero entries created during factorization. In one-dimensional problems consecutive numbering results in zero fill. In two- and three-dimensional problems fill can be considerable. Reconsider grid 3.20(b), the corresponding matrix in eq.(3.114) and factor ma-

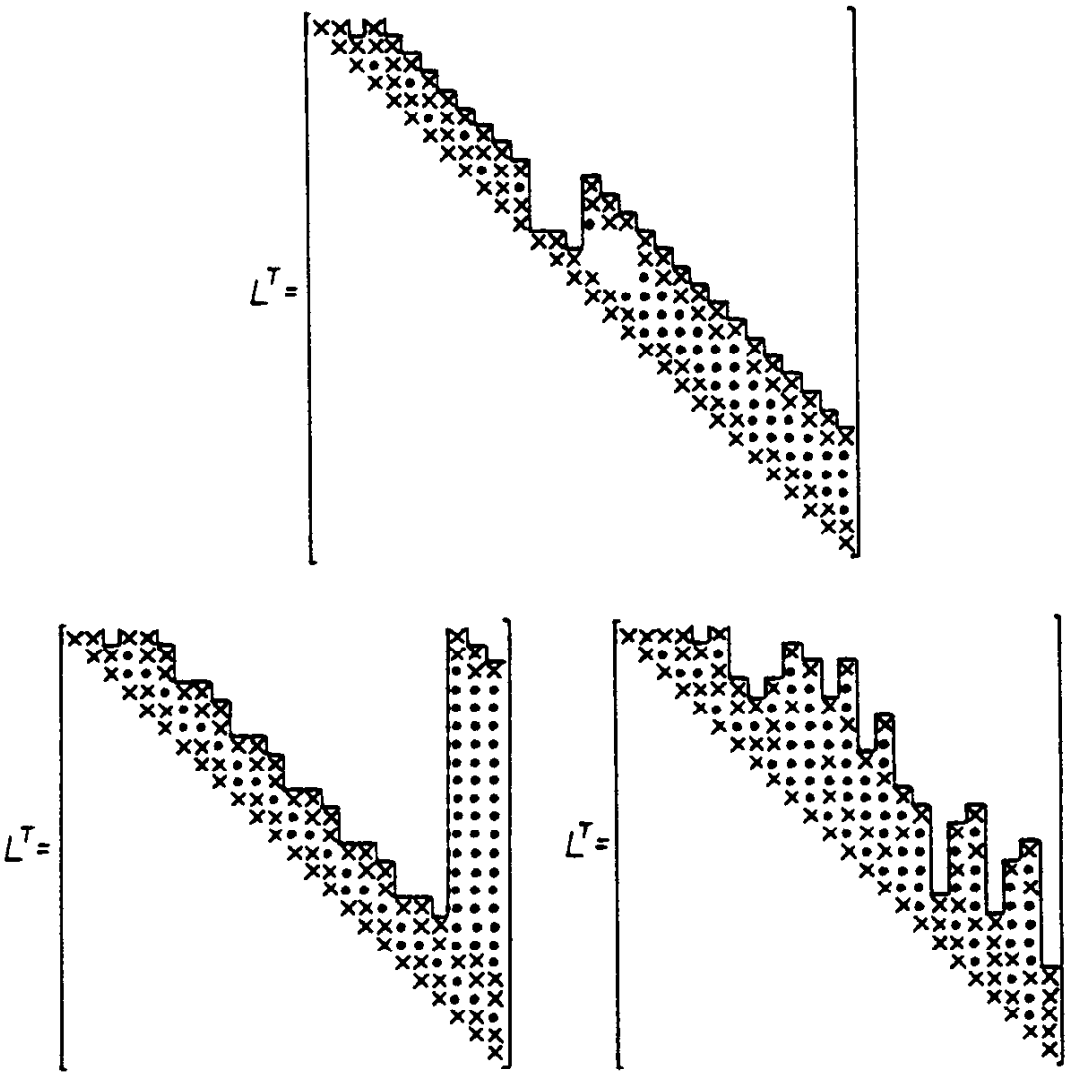


(3.118)

(3.119) (3.120)

trix L^T in eq.(3.116). We count 121 nonzero entries (marked $\times$ in eq.(3.116)) in the upper triangular part of K, and 125 newly created entries (marked $\bullet$) in L^T. Numbering along diagonals as in Fig. 3.20(c) reduces the fill as seen in eq.(3.117). Only 70 nonzero entries are created during the LL^T factorization of this K.

No polynomial-time algorithm exists for a minimum-fill numbering strategy but a simple heuristic rule is observed to regularly produce good results. It is the *minimum degree* algorithm. Nodes are first numbered in any way and matrix K is symbolically written to mark the zero and nonzero entries. Rows and columns are then symmetrically interchanged to have a first equation with the *least number of nonzero coefficients*. Usually there are several



(3.122)

(3.123) (3.124)

candidates for the first place and one among them is picked up arbitrarily. Node 1 is thus fixed, x_1 is eliminated from all equations below the first, and the left $n-1$ equations are searched again for the least number of nonzero coefficients in order to fix node 2. This is continued until all pivots are used up.

The minimum degree algorithm applied to K with the initial numbering of Fig. 3.20c produced the permuted node labels shown in Fig. 3.23, and L^T in eq.(3.125). The number of created nonzero entries is reduced from 70 to 62 but at the price of dispersing the nonzero entries of K all over the matrix.

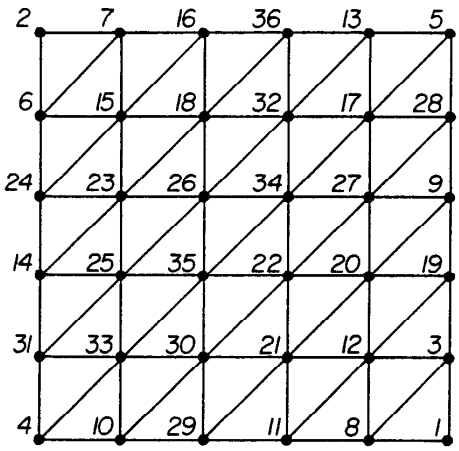
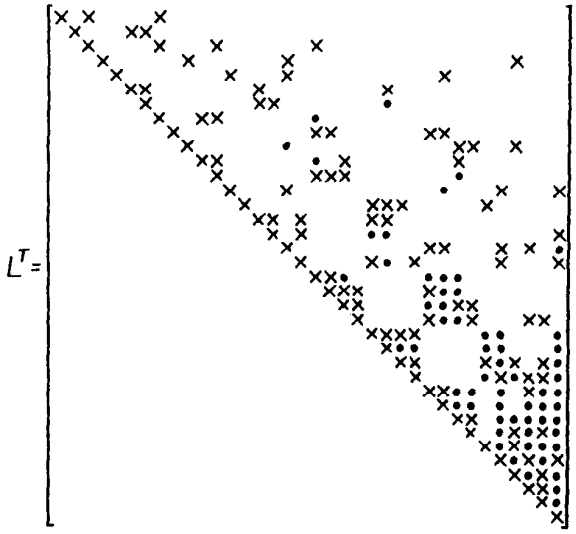


Fig. 3.23



(3.125)

Exercises

3.10.1. Number the nodes of the three grids (graphs) in Fig. 3.24

so as to have a stiffness matrix of a narrow band.

For a given numbering of the nodes define the *diameter*

$$\delta = \max |i - j|$$

of the graph as the greatest difference between any pair of *connected* nodes numbered i and j. The objective is to minimize δ over all possible numberings. Use the simple heuristic numbering strategy of returning to the connections of a labeled node as soon as possible.

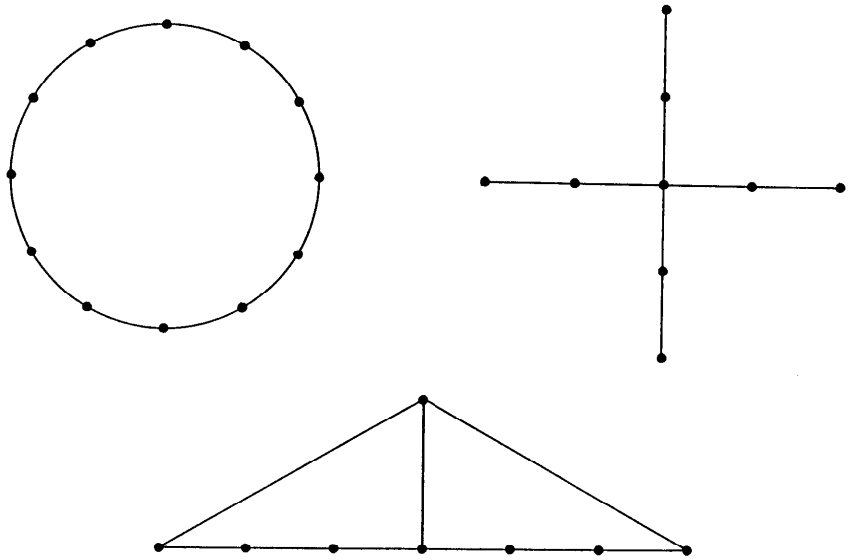


Fig. 3.24

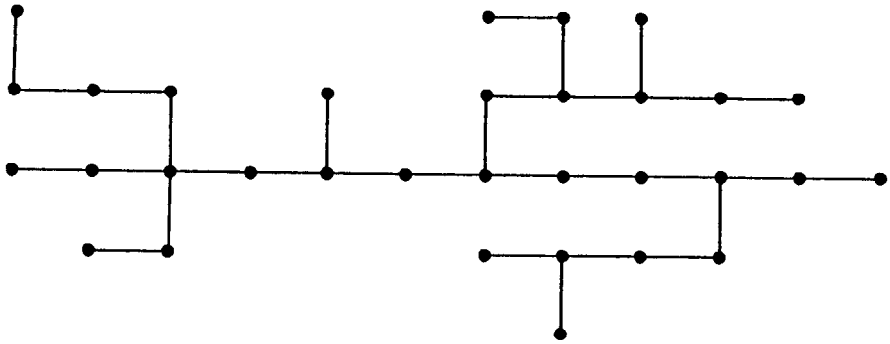


Fig. 3.25

3.10.2. The pipeline layout in Fig. 3.25 is in the language of graph theory a tree structure.

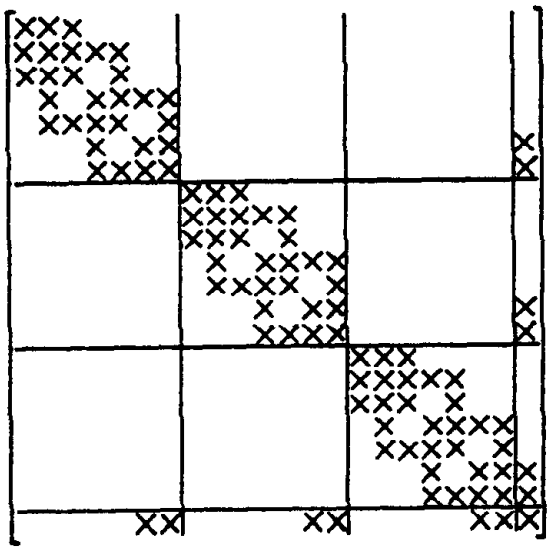
Number the nodes for a low δ. This example serves as a common challenge problem in the literature on band minimization algorithms.

3.11 Block algorithms

These are sparse *submatrix* storage and arithmetic schemes. An entry of K is accessed through a two-tier indexing; an outer for the submatrices and an inner for the elements of

each submatrix. A submatrix that remains zero throughout the factorization is ignored. In programs of different levels of complexity the nonzero submatrices are either considered dense or a sparse algorithm is individually applied to each one of them.

Engineers call this storage mode *substructuring*, as it is a natural choice for structures with repeated components held together at a few joints. Consider the tripod frame of Fig. 3.26 with three identical limbs tied at one point. The node-numbering system chosen in Fig. 3.26 is most sensible for this graph and it produces the stiffness matrix in eq. (3.126).

$K =$ 

(3.126)

Partitioned, the linear system $Ku = f$ for the tripod assumes the form

$$\begin{bmatrix} K_{11} & & & K_{14} \\ & K_{22} & & K_{24} \\ & & K_{33} & K_{34} \\ K_{14}^T & K_{24}^T & K_{34}^T & K_{44} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \tag{3.127}$$

where u_4 is the unknown at connecting joint 22. In terms of the submatrices

$$\begin{aligned} u_1 &= -K_{11}^{-1} K_{14} u_4 + K_{11}^{-1} f_1 \\ u_2 &= -K_{22}^{-1} K_{24} u_4 + K_{22}^{-1} f_2 \\ u_3 &= -K_{33}^{-1} K_{34} u_4 + K_{33}^{-1} f_3 \end{aligned} \tag{3.128}$$

$$K_{14}^T u_1 + K_{24}^T u_2 + K_{34}^T u_3 + K_{44} u_4 = f_4$$

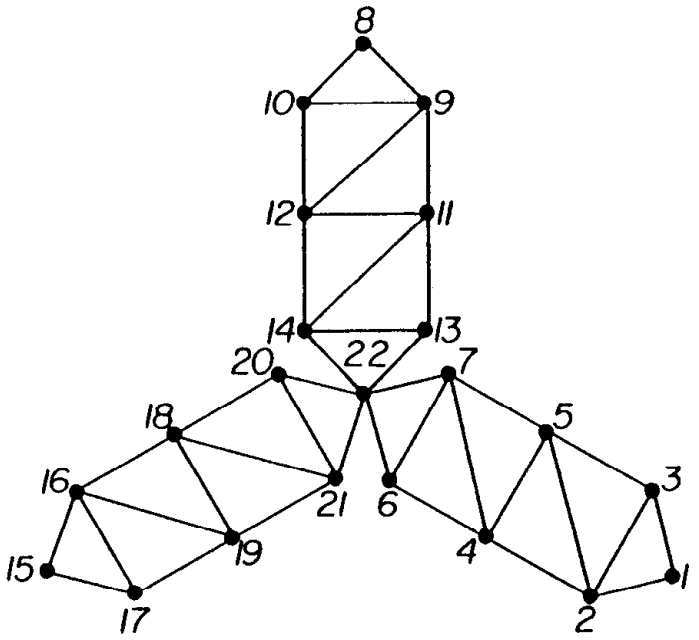


Fig. 3.26

entailing the solution of the six equations

$$\begin{array}{ll} K_{11}x_1 = K_{14} & K_{11}y_1 = f_1 \\ K_{22}x_2 = K_{24} & K_{22}y_2 = f_2 \\ K_{33}x_3 = K_{34} & K_{33}y_3 = f_3 \end{array} \tag{3.129}$$

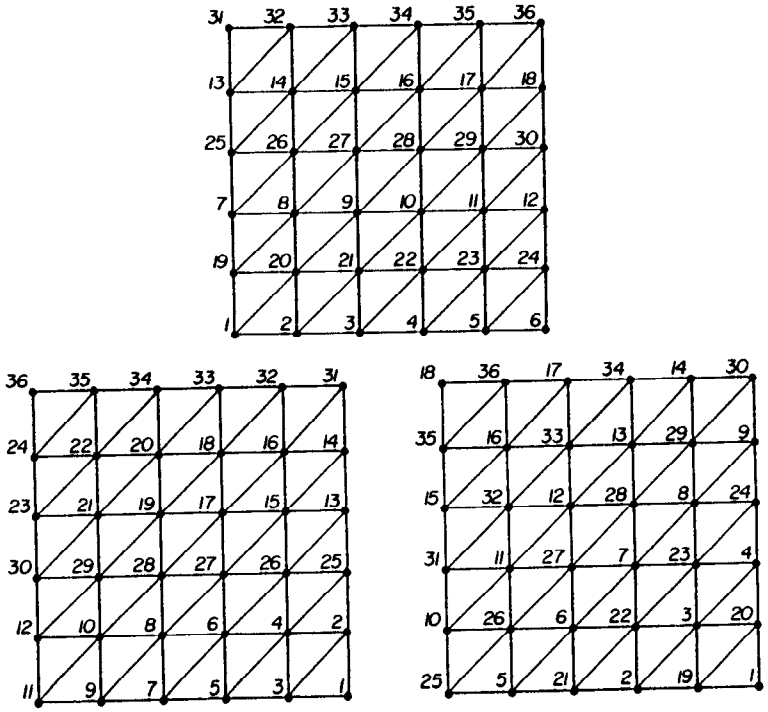
each with a 7×7 stiffness matrix. But for a repeating structure $K_{11} = K_{22} = K_{33}$, and only one substructure, or super-element, stiffness matrix needs to be set-up and factored. With systems (3.129) solved, u_1, u_2, u_3 are expressed in terms of u_4, and after their substitution into the fourth of eqs. (3.128) the equation is solved for the remaining unknown u_4.

Block-formed stiffness matrices can be created artificially with node-numbering systems that *separate* groups of node numbers. Three such examples are shown in Figs. 3.27(a), 3.27(b)

and 3.27(c), with the corresponding L^T factors written in eqs. (3.130),(3.131) and (3.132), respectively.

3.12 Arithmetic operations count

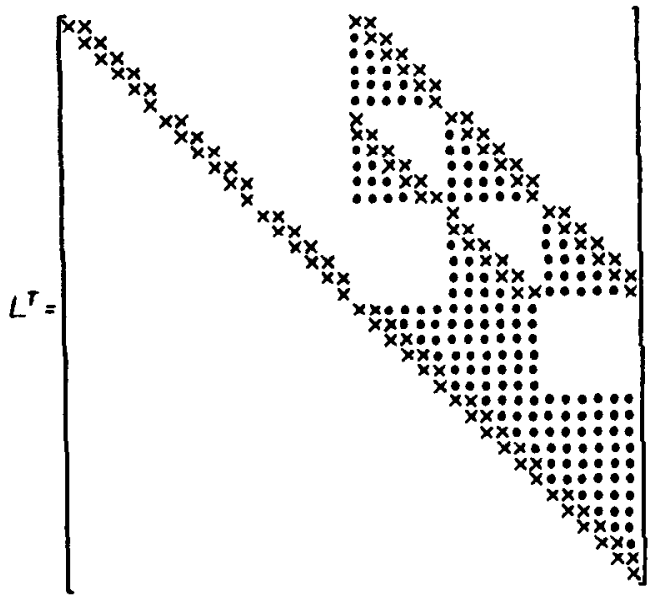
Simple yet reasonably realistic cost estimates are made here for the basic algorithms of computational matrix algebra in terms of matrix size and form and computer speed. Matrix



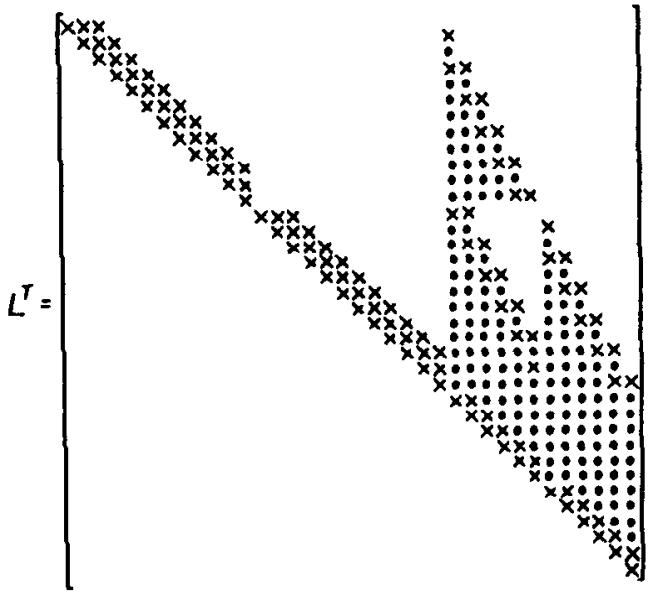
(a)

(b) (c)

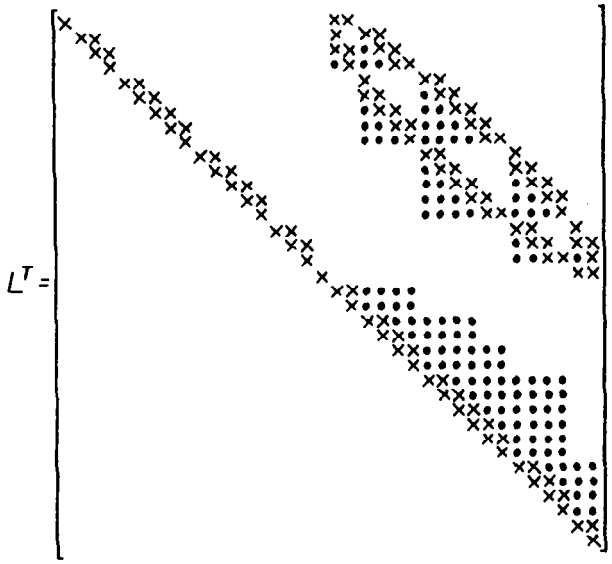
Fig. 3.27



(3.130)



(3.131)



(3.132)

arithmetic consists mainly of the repeated sequence of the retrieval of two entries from the computer storage, their multiplication, and the addition of the product to a stored floating-point number. Such a sequence we name an *operation*. Assuming that the entire matrix is in

random-access storage, one operation typically lasts 10^{-6} seconds on a mainframe computer and 10^{-4} seconds on a desktop computer. Discounting overhead, the operations count of an algorithm is proportional to the computing time and is directly translated into cost.

We commence with full matrices.

Theorem 3.3. *Let $A = A(n \times n)$ and $B = B(n \times n)$ be two dense matrices, $a = a(n \times 1)$ and $b = b(n \times 1)$ two vectors; and $L = L(n \times n)$ and $U = U(n \times n)$ dense lower- and upper-triangular matrices. The operations count for performing*

1. *$a^T b$ is n.*

2. *Ab is n^2.*

3. *AB is n^3.*

4. *LL^T is $\frac{1}{6}n(n+1)(n+2) = \frac{1}{6}n^3$ if $n >> 1$.*

5. *LU is $\frac{1}{6}n(n+1)(2n+1) = \frac{1}{3}n^3$ if $n >> 1$.*

Proof.

1. This statement is shorthand for $0 + a_1 b_1 + a_2 b_2 + \ldots + a_n b_n$ and to carry it out numerically requires n operations.

2. Formation of Ab entails the inner product of n pairs of $(n \times 1)$ vectors and hence the $nn = n^2$ operations.

3. Each entry of AB is the inner product of a row by a column, and there are n^2 entries in AB.

4. Since LL^T is symmetric, only the lower-triangular part of it need be computed; the entries above the diagonal are inserted symmetrically. Refer to eq. (3.133)

$$LL^T = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & & & \\ \times & \times & & \\ \times & \times & \times & \\ \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} \tag{3.133}$$

and grant that LL^T is computed columnwise. The first column of LL^T has n entries, each requiring one operation. The second column has $n - 1$ entries each requiring 2 operations.

The mth column has $n+1-m$ entries each requiring m operations. In all

$$\begin{aligned} 1n + 2(n-1) + 3(n-2) + \ldots + n &= \sum_{j=1}^{n} j(n-j+1) \\ &= (n+1)\sum_{j=1}^{n} j - \sum_{j=1}^{n} j^2 \end{aligned} \tag{3.134}$$

which with the summation formulas

$$1 + 2 + 3 + \ldots + n = \frac{1}{2}n(n+1) \quad , \quad 1^2 + 2^2 + 3^2 + \ldots + n^2 = \frac{1}{6}n(n+1)(2n+1) \tag{3.135}$$

yields $n(n+1)(n+2)/6$ operations.

4. No symmetry savings are available for the LU multiplication and all entries need be computed, requiring

$$\begin{aligned} 1(2n-1) + 2(2n-3) + 3(2n-5) + \ldots + n1 &= \sum_{j=1}^{n} j(2n-(2j-1)) \\ = (2n+1)\sum_{j=1}^{n} j - 2\sum_{j=1}^{n} j^2 &= \frac{1}{6}n(n+1)(2n+1) = \frac{1}{3}n^3 \end{aligned} \tag{3.136}$$

operations if $n >> 1$. End of proof.

Theorem 3.4. *Assume that in $Ax = f, A = A(n \times n)$ is dense, and $f = f(n \times 1)$. If the Gauss solution of the system is carried out without pivoting, then:*

1. Forward elimination requires

$$\frac{1}{6}n(n-1)(2n+5) \cong \frac{1}{3}n^3 \tag{3.137}$$

operations if A is unsymmetric, but only

$$\frac{1}{6}n(n-1)(n+7) \cong \frac{1}{6}n^3 \tag{3.138}$$

operations if A is symmetric.

2. Back substitution requires

$$\frac{1}{2}n(n+1) \cong \frac{1}{2}n^2 \tag{3.139}$$

operations.

3. Inversion and multiplication by the right-hand side to produce $x = A^{-1}f$ requires

$$n^2(n+1) \cong n^3 \tag{3.140}$$

operations.

Proof.

1. Elimination of x_1 from each equation below the first requires n operations on the matrix and one operation on the right-hand side, altogether $(n+1)(n-1)$ operations. Zeroes created during elimination are ignored and hence elimination of x_2 from all equations below the second requires $n(n-2)$ operations. The entire forward elimination process consists of

$$0 \cdot 2 + 1 \cdot 3 + 2 \cdot 4 + \ldots + (j-1)(j+1) + \ldots + (n-1)(n+1) \tag{3.141}$$

operations, summed up to yield

$$\sum_{j=1}^{n}(j^2 - 1) = \frac{1}{6}n(n-1)(2n+5) \cong \frac{1}{3}n^3 \tag{3.142}$$

operations.

Now let A be symmetric and suppose that x_1 is eliminated from all equations below the first so that the newly created equivalent system $A'x = f'$ is with

$$A' = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & \times & \times \end{bmatrix}. \tag{3.143}$$

The lower portion of A' is symmetric and as a result only a triangular part of A need be stored and modified by the Gauss algorithm. Elimination of x_1 requires $n-1$ divisions of the pivot, $\frac{1}{2}n(n-1)$ operations on the matrix, and $n-1$ operations on the right-hand side vector. Altogether

$$\sum_{j=1}^{n}\left(\frac{1}{2}j(j-1) + 2(j-1)\right) = \frac{1}{6}n(n-1)(n+7) \tag{3.144}$$

operations.

2. In back substitution the jth equation is divided by the jth pivot, and $j-1$ operations are performed on the right-hand side. Hence the total of

$$\sum_{j=1}^{n} j = \frac{1}{2}n(n+1) \tag{3.145}$$

operations.

3. Solution of $Ax = f$ through $x = A^{-1}f$ calls for the inversion of A, and a vector matrix multiplication. Inversion is what we have to look at. We write $AA^{-1} = I$ and perform forward elimination to transform it into

$$\begin{bmatrix} \times & \times & \times & \times \\ & \times & \times & \times \\ & & \times & \times \\ & & & \times \end{bmatrix} A^{-1} = \begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & \times & 1 & \\ \times & \times & \times & 1 \end{bmatrix}. \tag{3.146}$$

Discounting multiplications by 0 and 1, we set out to evaluate the work done on the right-hand side matrix. Because the first column of I has only a 1 and $(n-1)$ zeros, no work is spent on the first column of the lower-triangular matrix. Formation of the second column requires work on the first column only.

The total work to create the lower-triangular right-hand side matrix consists accordingly of

$$\sum_{j=1}^{n}(j-1)(n-j) = \frac{1}{6}n(n-1)(n-2) \tag{3.147}$$

operations. From part 2 of this theorem we have that the number of arithmetic operations needed to create the upper triangular matrix at the left-hand side is

$$\sum_{j=1}^{n} j(j-1) = \frac{n}{3}(n^2-1) \tag{3.148}$$

and hence the total of

$$\sum_{j=1}^{n} nj - n = \frac{1}{2}n^2(n-1) \tag{3.149}$$

operations.

Back substitution starts by making the current pivot 1 through division. No work is needed on the left-hand side matrix but the right-hand side requires

$$n\sum_{j=1}^{n}(n-j+1) = \frac{1}{2}n^2(n+1) \tag{3.150}$$

operations. Altogether, approximately n^3 operations are needed to separately write A^{-1} and n^2 operations for the product $A^{-1}f$. End of proof.

Theorem 3.5. *When n is large:*

1. *The LU, $L_{ii} = 1$, factorization of $A = A(n \times n)$ requires $\frac{1}{3}n^3$ operations.*
2. *The symmetric LL^T factorization of A requires $\frac{1}{6}n^3$ operations plus n square roots.*

Proof. The LU factorization is accomplished by an alternate computation of the columns of U and L. We verify that

$$\frac{1}{2}\sum_{j=1}^{n} j(j-1) \text{ and } \sum_{j=1}^{n} j(n-j) \tag{3.151}$$

operations are needed to compute the columns of U and L, respectively. In sum,

$$\frac{1}{3}n(n^2-1) \cong \frac{1}{3}n^3 \tag{3.152}$$

operations are required for the complete factorization.

To create L in $LL^T = A$ we need

$$\sum_{j=1}^{n} (j(n-j) + (j-1)) = \frac{1}{6}n(n-1)(n+4) \tag{3.153}$$

operations, plus n square roots to determine L_{ii}. End of proof.

Theorems 3.4 and 3.5 make it emphatically clear that solution of $Ax = f$ by forming the inverse A^{-1} is not worthwhile even with several different right-hand sides. Inversion of A requires n^3 operations but the LU factorization only $n^3/3$ operations. Writing $Ax = f$ as $LUx = f$ and repeatedly solving

$$Lx' = f,\ Ux = x' \tag{3.154}$$

calls for one factorization for any number of right-hand sides, and two back substitutions for any specific f.

In any event, inversion of the large sparse finite difference matrices is utterly out of the question as no place could be found to accommodate the full inverse, even for systems that

can otherwise be stored and Gauss-solved by sparse codes. For sparse systems of linear equations with a positive definite and symmetric matrix and several right-hand side vectors the LL^T factorization of the matrix and the successive solution of $Lx' = f$ and $L^T x = x'$ is the only practical thing to do.

We turn now to the practically important band matrices.

Theorem 3.6. *Let $A = A(n \times n)$ be a symmetric positive definite band matrix of bandwidth $2k + 1$.*

1. The number of operations needed to factor A into LL^T is

$$\frac{1}{2}nk(k+3) - \frac{1}{3}k(k+1)(k+2) \tag{3.155}$$

plus n square roots.

2. The number of operations needed to solve $Ax = f$, given L is

$$(k+1)(2n-k). \tag{3.156}$$

Proof. Consider

$$LL^T = \begin{array}{c} \begin{matrix} k & & n-2k & & k \end{matrix} \\ \left[\begin{array}{cc|ccc|cc} \times & & & & & & \\ \times & \times & & & & & \\ \hline \times & \times & \times & & & & \\ & \times & \times & \times & & & \\ & & \times & \times & \times & & \\ \hline & & & \times & \times & \times & \\ & & & & \times & \times & \times \end{array}\right] \left[\begin{array}{cc|ccc|cc} \times & \times & \times & & & & \\ & \times & \times & \times & & & \\ \hline & & \times & \times & \times & & \\ & & & \times & \times & \times & \\ & & & & \times & \times & \\ \hline & & & & & \times & \times \\ & & & & & & \times \end{array}\right] \end{array} \begin{matrix} k \\ \\ n-2k \\ \\ k \end{matrix} . \tag{3.157}$$

The number of operations needed to factor A is that required to carry out the product LL^T, except for the n operations for the diagonals that are square roots. The work involved in writing the first and last k columns of L amounts to

$$2\sum_{j=1}^{k}\left(j(k+\frac{3}{2}) - \frac{1}{2}j^2\right) = \frac{2}{3}k(k+1)(k+2) \tag{3.158}$$

operations. All columns from the $(k+1)$th to the $(n-k)$th require the same $(k+1)(k+2)/2$ operations, and hence the work for the $n-2k$ columns is

$$\frac{1}{2}(k+1)(k+2)(n-2k) \tag{3.159}$$

operations. Adding the work for all columns minus the n diagonal operations we arrive at the expression in the theorem.

When $n >> 1$ and $1 < k << n$ the factorization requires close to

$$\frac{1}{2}nk^2 \tag{3.160}$$

operations. Back substitution requires $(n-k)(1+k)$ operations for the last $n-k$ columns, and $\frac{1}{2}k(k+1)$ operations for the first k columns. There are two back substitutions for each right-hand side and expression (3.156) in the theorem is recovered.

When $n >> 1$ and $1 < k << n$ back substitution requires approximately $2kn$ operations. End of proof.

In light of Theorem 3.6 we can appreciate the quest for a narrow bandwidth. Symmetric storage of a band matrix of bandwidth $2k+1$ calls for only

$$\frac{1}{2}(k+1)(2n-k) \cong kn \tag{3.161}$$

locations and its factorization cost is proportional to n and merely k^2.

When k is small the cost of computing the n square roots needed in LL^T becomes relatively heavy. An LDL^T factorization that requires no square roots could be cheaper for such narrow band matrices.

A square grid of plane finite differences with m nodes per side gives rise to a linear system with a coefficient matrix of order $n = m^2$, of half bandwidth $k = m$. Storage of the matrix takes up about m^3 locations and its LL^T factorization requires $\frac{1}{2}m^4$ operations. Say $m = 25,\ n = 625$ and $k = 25$. This requires some $16 \cdot 10^3$ storage locations and $20 \cdot 10^4$ operations. At the rate of 10^{-6} seconds per operation the factorization is accomplished in 0.2 seconds.

Three-dimensional problems are considerably more expensive. A cube with m nodes per side gives rise to a coefficient matrix of order $n = m^3$, of half bandwidth $k = m^2$. Some m^5

storage locations and $\frac{1}{2}m^7$ operations are consumed in the factorization. If $m = 25$ and the cost per operation is 10^{-6} seconds, then factorization languishes for over 3000 seconds or 50 minutes.

Exercises

3.12.3. The serious student should clock the computer he is using to know the time it takes to perform an arithmetical operation. A simple program to carry out the inner product $a^T b$ of two long arrays will furnish a realistic estimate for the computer speed.