Introduction to Finite Elements
(Matrix Methods)

for

ME 345 Modeling and Simulation

Spring 2005

Professor Frank Fisher

Notes: These notes are borrowed extensively from a Finite Elements primer written by Professors Belytschko and Brinson (January 1995) and revised for Engineering Analysis 2 by Professors Moran and Krishnaswamy (February 1997) at Northwestern University.

TABLE OF CONTENTS

INTRODUCTION TO STATIC, LINEAR STRESS ANALYSIS BY FINITE ELEMENTS

Introduction.

One of the most commonly used methods of stress analysis is the finite element method—a matrix based method of solving problems which was developed for structural analysis of aircraft and later recognized as a versatile tool for which a rigorous mathematical foundation could be laid. It is currently one of the most-widely used analysis tools in all aspects of engineering and sicence. Almost any manufactured part or system, including components such as gears, cams, and suspensions, to complete systems, such as automobiles and aircraft, are analyzed by finite elements to ensure that they have the durability and reliability required and that they meet performance requirements. Finite element methods are also used in biomedical application such as the design of prostheses, modeling the nonlinear dyanamics of the human heart or analysing gait, for example. Finite element programs are also used for many other types of analysis and design: fluid dynamics, heat transfer, electromagnetic fields are some examples.

A major reason for the popularity of the finite element method is that a single program can perform the analysis of almost any component or structure: the geometry of the object and its loads are defined by *mesh* data and the program then sets up the governing equations in a straightforward manner. Examples of finite element meshes for some industrial problems are shown in Figures 1 to 4. Finite element analyses today are usually performed by general purpose programs which can do a large variety of analyses: stress analysis, vibration analyses, optimum design are a sample of some of the functions of general purpose programs. Furthermore, many of these programs can perform both linear and nonlinear analysis.

The type of stress analysis which will be taught in this course is linear, static stress analysis. A large part of stress analysis in industry is linear, static analysis. Most stress analysis taught at the undergraduate level is linear and static. Nonlinear analysis is usually used only for evaluating the performance under extreme environments.

Fundamental assumptions.

The assumptions which are made in *linear* static stress analysis are:

1. The displacements of the structure are small compared to its dimensions.
2. The material is linear and elastic.
3. The response of the structure is static (or steady-state).

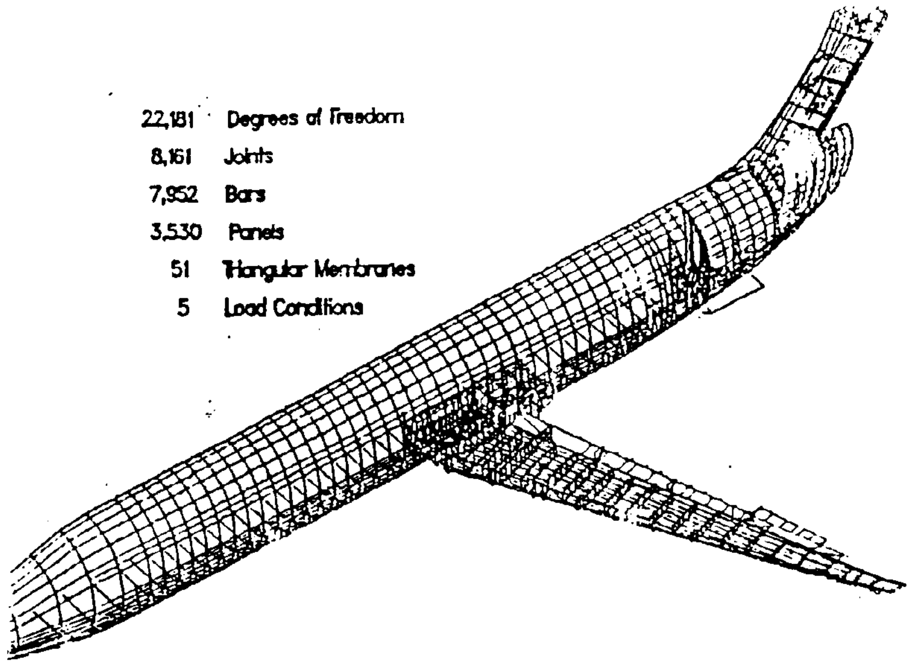


Fig. 1. Finite element model of a DC-9 stretch jet.

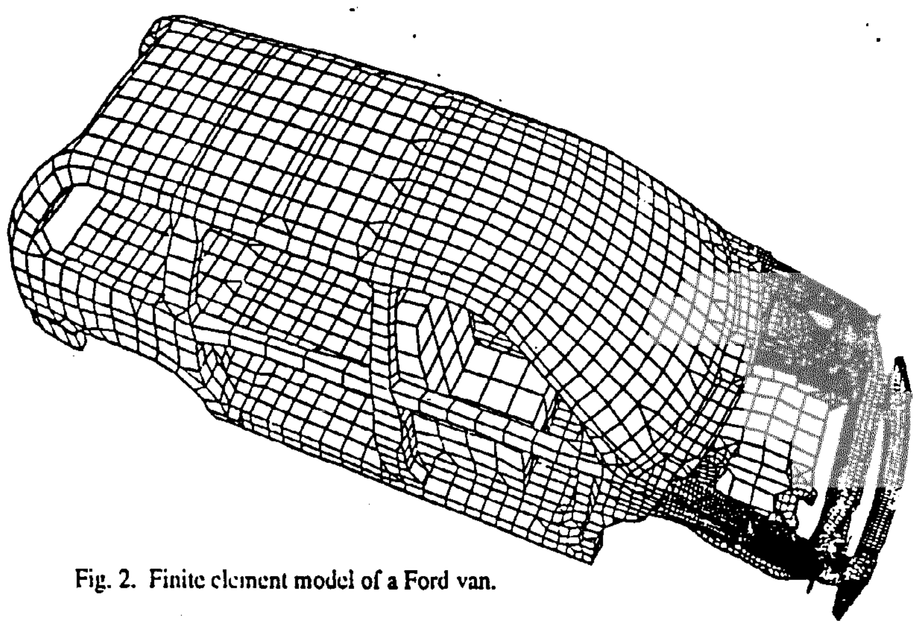


Fig. 2. Finite element model of a Ford van.

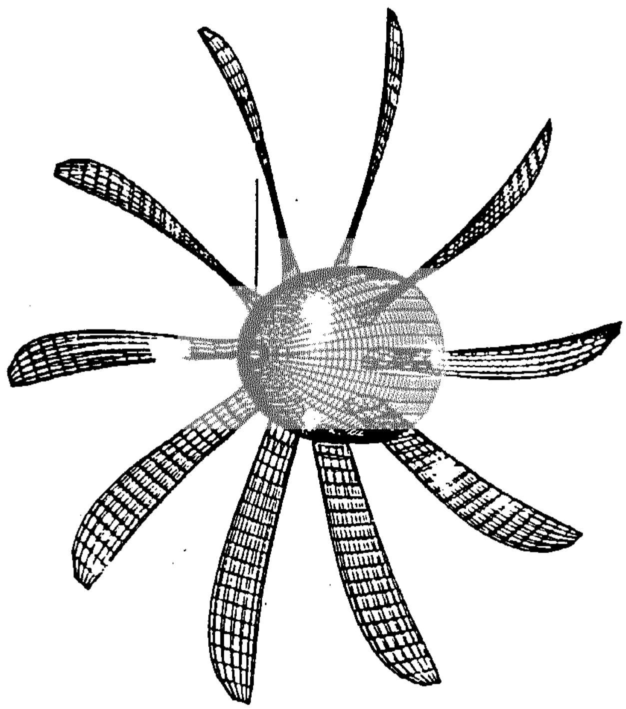


Fig. 3. Finite element model of a truboprop fan.

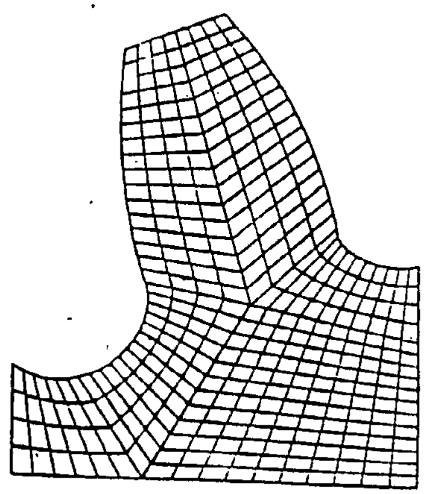


Fig. 4. Finite element model of a gear tooth.

The first assumption implies the geometry of the structure does not change appreciably due to the application of the loads. Almost all stress analyses considered in undergraduate strength of materials courses are small displacement problems.

The assumption of a linear, elastic material implies two characteristics:

i. the relation between the stress and strain, and hence the relation between the forces and displacements, is *linear*.

ii. the strains are reversible, that is, when the loads are removed, the strains and displacements return to their original values, which are usually zero (this is called *elastic* behavior).

Some elastic materials are not linear; an example is rubber, for which the strains are completely reversible (a rubber band will return to its original length after stretching), but the stress is not a linear function of the strain.

Most metals are linear, elastic when the stresses remain below the yield-point, or elastic limit. Above the yield point, the metal sustains plastic strains which are not reversible, which means they will be deformed after release of the load. As an example, consider the bending of a paper clip. For small deformations, the clip will revert to its original shape after it is deformed, but if the deformations are large enough, the clip will be permanently deformed, which is evidence of irreversible, plastic deformation.

The assumption of static behavior implies that dynamic or inertial effects are negligible because of the slow rate at which the loads are applied.

When the displacements of a structure are large or the material behavior is nonlinear, *nonlinear analysis* must be used. Nonlinear analysis will not be studied in this course.

Basic equations of linear stress analysis.

Any analysis of stresses in a static body must satisfy the following:

1. equilibrium
2. stress-strain law
3. compatibility and strain-displacement equations

Equilibrium requires that the sum of the forces and the moments vanish at all points of the structure. This is a consequence of Newton's second law of motion which states that the resultant force acting on a body is equal to the rate of change of linear momentum (recall that linear momentum is equal to mass times velocity) and the resultant moment is equal to the rate of change of angular momentum (equal to moment of inertia times angular velocity). See Bedford and Fowler *Dynamics* Chapters 2 and 3 and Bedford and Fowler *Statics* Chapters 2 and 3 for a more detailed discussion of Newton's Laws.

$$\sum \mathbf{F} = \frac{d}{dt}(m\mathbf{v})$$
$$\sum \mathbf{M} = \frac{d}{dt}(I\omega) \tag{1.1}$$

Thus, for a body which is at rest (or moving at a constant velocity) the sum of the forces and the sum of the moments of the forces acting on it must vanish. In this case the equilibrium equations are written as:

Equilibrium

$$\sum \mathbf{F} = 0$$
$$\sum \mathbf{M} = 0 \tag{1.2}$$

Stress Strain Law

The stress-strain law depends on the material. For axial stretching of a rod, we saw that:

$$\sigma = E\varepsilon \tag{1.3a}$$

where σ is the stress, E is Young's modulus and ε is the strain. This is called Hooke's law. Note that Young's modulus has dimensions of stress (N / m^2).

For a spring, the force-elongation relation is analogous to the stress strain law for a material, i.e,

$$f = k\delta \tag{1.3b}$$

where f is the force in the spring, k is the spring constant (which depends on the spring geometry and coil arrangement and δ is the elongation of the spring. Note that the spring constant has dimensions of force per unit length (N / m).

Compatibility

Compatibility requires that the displacements be continuous everywhere in the body. In later courses, you will see how to represent displacements as a function of position and will then be in a position to describe compatibility in a more general sense involving restrictions on strain and on the continuity of displacements. For now, it suffices for us to think of compatibility as the restriction that the displacements be continuous. In particular, if two bodies (or parts of a single body) are connected at a point, each part of the body experiences the same displacement at that point. We will also require that the bodies or parts of the body not break apart. These conditions imply that there are no gaps or overlaps in the body.

Element Stiffness Matrix for1D Spring.

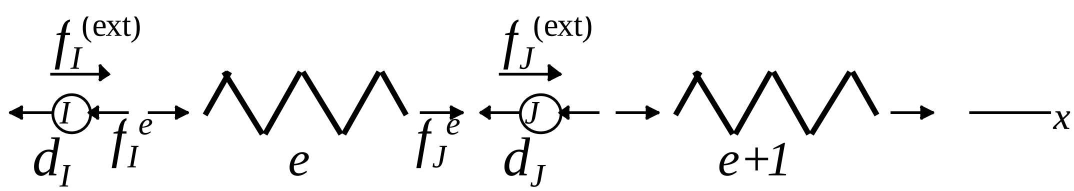
The element stiffness relates the element nodal internal forces $\mathbf{f}_e$ (the superscript "int" has been omitted) to the element nodal displacements $\mathbf{d}_e$ by

$$\mathbf{f}_e = \mathbf{K}_e \mathbf{d}_e \tag{1.4}$$

The subscripts “e” indicate that the matrices pertain to an element. When it is clear that the matrices are related to elements, the subscripts “e” are often dropped. Note that nodal displacements are denoted by $\mathbf{d}_e$

The element stiffness matrix is derived by requiring equilibrium, the spring law (which is the counterpart of a stress-strain law), and compatibility to be satisfied on an element level.

The element nomenclature is defined below

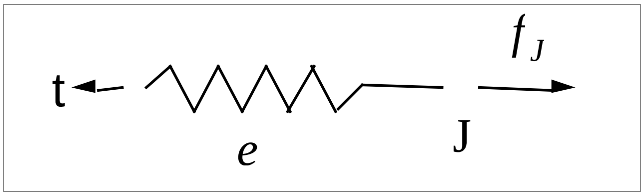


$$\mathbf{f}_e = \begin{Bmatrix} f_I \\ f_J \end{Bmatrix} \quad \mathbf{d}_e = \begin{Bmatrix} d_I \\ d_J \end{Bmatrix}$$

The spring law states that the tension t in the spring is given by

$$t = k\delta \tag{1.5}$$

where k is the spring constant and δ is the elongation of the spring.



From a free-body diagram, we can see that equilibrium of the part of element *e* shown gives (the superscript denoting the element number is dropped for convenience)

$$f_J = t \tag{1.6}$$

and equilibrium of the whole element gives

$$f_I = -f_J \tag{1.7}$$

By the definition of elongation

$$\delta = d_J - d_I \tag{1.8}$$

Combining (1.6), (1.5) and (1.8) gives

$$f_J = t = k\delta = k(d_J - d_I) \tag{1.9}$$

and (1.7) and (1.9) give

$$f_I = -f_J = -k(d_J - d_I) = k(d_I - d_J) \tag{1.10}$$

Writing (1.9) and (1.10) in matrix form

$$\begin{Bmatrix} f_I \\ f_J \end{Bmatrix} = k \underbrace{\begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}}_{\mathbf{K_e}} \begin{Bmatrix} d_I \\ d_J \end{Bmatrix} \tag{1.11}$$

The above equation gives the element stiffness matrix $\mathbf{K}_e$ for the 1D spring. Note that $\mathbf{K}_e = \mathbf{K}_e^T$, i.e. $\mathbf{K}_e$ is symmetric.

Remark 1. The nodal forces and nodal displacements are always defined in finite element methods so that they are positive in the positive coordinate direction. This is crucial for easy assemby of the global equations, as will be seen later.

Remark 2. The work done by the forces is proportional to force times displacement. This work is equal to the energy stored in the stretched body. The nodal forces and displacements are arranged so that $\mathbf{d}^T\mathbf{f}$ is proportional to the work done; otherwise $\mathbf{K}_e$ is not symmetric.

Assembly of global equations in one dimension.

One of the most attractive features of the finite element method is its ability to treat a large variety of geometries, loadings and boundary conditions in a single program. This versatility arises from the fact that a finite element program does not incorporate a specific geometry or loading in its coding, but generates the global equations from the element equations by assembling the element equations according to the input data. In this section, the assembly operation is described and illustrated for one dimensional problems.

The assembly operation is designed to meet the basic requirements of a solution:

i. compatibility
ii. equilibrium
iii. stress-strain law

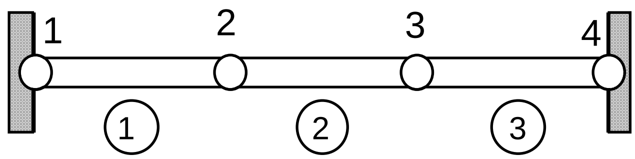
The assembly operation will now be described for a 3 element mesh for the problem shown in Table 1A. In general purpose programs, the code is written so that the procedure is independent of the node numbering. When elaborate meshes are generated for complicated industrial components (Figures 1-4) the resulting node numbering is usually haphazard and a code which can account for this is required. For the one-dimensional spring and rod codes used here, the node numbering is taken to be sequential, as shown in Table 1A. This makes for a simpler code, and also for an easier illustration of the method. The element numbers are enclosed by circles. The element stiffness matrices are shown below the mesh. Symbols are used for the elements of the stiffness matrix so that you can see exactly where each term of the element stiffness matrix goes in the assembly procedure. The element numbers are indicated by superscripts in parenthesis or circled superscripts and the node numbers I and J of each element are indicated. The element applies forces only through those nodes which are connected to the element.

The assembly is first described using a technique called *the augmented matrix procedure*, which clarifies how the requirements of a solution are imposed. The first operation in the assembly by the augmented matrix procedure is indicated in Table 1B. Here, each element stiffness matrix is first augmented by adding zeros to any rows or columns which are not connected to that element. This accounts for the fact that the element only applies forces to those nodes which are connected to the element and these forces only depend on the displacements of the element nodes.

For example, element 1 is not connected to nodes 3 or 4, so it makes no direct contributions to the forces at nodes 3 and 4 and consequently rows 3 and 4 of the augmented matrix for element 1 are all zeros. Similarly, the displacements of nodes 3 and 4 have no effect on the response of element 1, so these columns are zero. Note that the elements of the stiffness matrix are arranged according to the node numbers: Node I of element 1 is node 1 of the mesh, k_{11} becomes the (1,1) entry of the augmented matrix, and k_{21} becomes the (2,1) entry of the augmented matrix etc.

For element 2, rows 1 and 4 and columns 1 and 4 become zeros because element 2 is not connected to those nodes. k_{11} becomes the (2,2) entry of the augmented matrix of element 2 since it relates the effect of the displacement of node 2 on the force on node 2, k_{12} becomes the (2,3) entry since it relates the force at node 2 to the displacement of node 3, etc.

TABLE 1A



element 1

$$\begin{Bmatrix} f_I^{(1)} \\ f_J^{(1)} \end{Bmatrix}^{int} = \underbrace{\begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} \\ k_{21}^{(1)} & k_{22}^{(1)} \end{bmatrix}}_{\mathbf{K^{(1)}}} \begin{Bmatrix} d_I^{(1)} \\ d_J^{(1)} \end{Bmatrix} \qquad \begin{matrix} I = 1 \\ J = 2 \end{matrix}$$

element 2

$$\begin{Bmatrix} f_I^{(2)} \\ f_J^{(2)} \end{Bmatrix}^{int} = \underbrace{\begin{bmatrix} k_{11}^{(2)} & k_{12}^{(2)} \\ k_{21}^{(2)} & k_{22}^{(2)} \end{bmatrix}}_{\mathbf{K^{(2)}}} \begin{Bmatrix} d_I^{(2)} \\ d_J^{(2)} \end{Bmatrix} \qquad \begin{matrix} I = 2 \\ J = 3 \end{matrix}$$

element 3

$$\begin{Bmatrix} f_I^{(3)} \\ f_J^{(3)} \end{Bmatrix}^{int} = \underbrace{\begin{bmatrix} k_{11}^{(3)} & k_{12}^{(3)} \\ k_{21}^{(3)} & k_{22}^{(3)} \end{bmatrix}}_{\mathbf{K^{(3)}}} \begin{Bmatrix} d_I^{(3)} \\ d_J^{(3)} \end{Bmatrix} \qquad \begin{matrix} I = 3 \\ J = 4 \end{matrix}$$

TABLE 1-B
AUGMENTED MATRICES

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{(1)\text{int}} = \begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & 0 & 0 \\ k_{21}^{(1)} & k_{22}^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix}^{(1)}$$

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{(2)\text{int}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_{11}^{(2)} & k_{12}^{(2)} & 0 \\ 0 & k_{21}^{(2)} & k_{22}^{(2)} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix}^{(2)}$$

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{(3)\text{int}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_{11}^{(3)} & k_{12}^{(3)} \\ 0 & 0 & k_{21}^{(3)} & k_{22}^{(3)} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix}^{(3)}$$

TABLE 1C
EQUILIBRIUM

Equilibrium at the nodes is given by

$$\sum \mathbf{F} = 0$$

$$\mathbf{f}^{(\mathrm{ext})} - \mathbf{f}^{(\mathrm{int})} = 0$$

which can be written as

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{ext} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{\mathrm{int}}$$

Expanding the right hand side of the above gives

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{ext} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{(1)\mathrm{int}} + \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{(2)\mathrm{int}} + \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{(3)\mathrm{int}}$$

SUBSTITUTE IN AUGMENTED STIFFNESS EQNS
AND ENFORCE COMPATIBILITY : $\mathbf{d} = \mathbf{d}^{(1)} = \mathbf{d}^{(2)} = \mathbf{d}^{(3)}$

$$\begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{ext} = \left(\begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & 0 & 0 \\ k_{21}^{(1)} & k_{22}^{(1)} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & k_{11}^{(2)} & k_{12}^{(2)} & 0 \\ 0 & k_{21}^{(2)} & k_{22}^{(2)} & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & k_{11}^{(3)} & k_{12}^{(3)} \\ 0 & 0 & k_{21}^{(3)} & k_{22}^{(3)} \end{bmatrix} \right) \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix}$$

Table 1C then represents the imposition of equilibrium requirements: the external forces at the nodes $\mathbf{f}^{ext}$ equals the internal forces $\mathbf{f}^{int}$ the sum of the contributions of all elements to the internal forces.

The element internal forces are then expressed in terms of the element nodal displacements via the augmented element stiffness matrices. Compatibility is enforced by requiring the nodal displacements for all elements to be the same, as shown in the bottom line of Table 1C. The stress-strain law has already been incorporated in the element stiffness matrices. We then sum the augmented element stiffnesses to obtain the total stiffness matrix given in Table 1D. Note that this matrix is symmetric. The total stiffness matrix is sometime called the global stiffness (or global system) matrix.

TABLE 1D

ASSEMBLED MATRIX (sum the matrices in parenthesis in 1C)

$$\begin{bmatrix} k_{11}^{(1)} & k_{12}^{(1)} & 0 & 0 \\ k_{21}^{(1)} & k_{22}^{(1)} + k_{11}^{(2)} & k_{12}^{(2)} & 0 \\ 0 & k_{21}^{(2)} & k_{22}^{(2)} + k_{11}^{(3)} & k_{12}^{(3)} \\ 0 & 0 & k_{21}^{(3)} & k_{22}^{(3)} \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix} = \begin{Bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{Bmatrix}^{ext}$$

The total stiffness given in Table 1D does not take into account the support conditions. In examining the model, it can be seen that f_1^{ext} and f_4^{ext} are not known prior to solving the system of equations since they are reaction forces. In this example the structure is statically indeterminate, so f_1^{ext} and f_4^{ext} are unknown before the solution is obtained. Even, in a statically determinate structure, the reaction forces would be considered unknowns in this computerized appraoch (sometimes called the stiffness method) since a general computer algorithm does not make any distinction between statically determinate and statically indeterminate problems. Note that we're not just using the equations of statics (equilibrium) here; rather, we use equilibrium, stress-strain law and compatibility—thus we can solve statically determinate and indeterminate problems without any special coding techniqes for one or the other.

Since the reactions (external forces supplied by the supports) f_1^{ext} and f_4^{ext} are not known, the first and fourth equations in Table 1D cannot be used in the form shown—in a solvable set of linear equations, the right hand sides must be known. To circumvent this difficulty, equations 1 and 4 are removed from the system. This leaves 2 equations in 4 unknowns, which is unsolvable. However, the displacement corresponding to each unknown reaction force is known, and in fact is zero (In more general treatments we can also account for non-zero prescribed displacements). Therefore, the first and fourth columns of the stiffness matrix, which multiply these zero displacements, can also be eliminated as shown in Table 1E.

TABLE 1E

Enforce Constraints (support conditions): $d_1=0$, $d_4=0$; this is achieved by deleting rows 1 and 4 and columns 1 and 4 of the above.

$$\begin{bmatrix} k_{22}^{(1)}+k_{11}^{(2)} & k_{12}^{(2)} \\ k_{21}^{(2)} & k_{22}^{(2)}+k_{11}^{(3)} \end{bmatrix} \begin{Bmatrix} d_2 \\ d_3 \end{Bmatrix} = \begin{Bmatrix} f_2 \\ f_3 \end{Bmatrix}^{ext}$$

After elimination of the rows and columns corresponding to the support nodes, the 2x2 system of equations shown in Table 1E remains. This matrix is also symmetric. The right hand sides of this system of linear algebraic equations are the known external forces. Solving the equations provides the displacements d_2 and d_3, from which the springs elongations and internal forces in the elements may be found.

An alternative way of imposing the constraints is shown in Fig. 1F. This procedure is more suited to a computer program because it eliminates the extensive bookkeeping that is necessary whenever rows and columns are eliminated. Here, the rows and columns are associated with a constraint are zeroed, a nonzero number (usually a 1) is placed on the diagonal, and the right hand side of the equation is replaced by zero. In effect, the procedure inserts the trivial equation that the displacement is zero at any node where this constraint applies.

TABLE 1F

ALTERNATIVE FINAL EQUATIONS

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & k_{22}^{(1)}+k_{11}^{(2)} & k_{12}^{(2)} & 0 \\ 0 & k_{21}^{(2)} & k_{22}^{(2)}+k_{11}^{(3)} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} d_1 \\ d_2 \\ d_3 \\ d_4 \end{Bmatrix} = \begin{Bmatrix} 0 \\ f_2 \\ f_3 \\ 0 \end{Bmatrix}^{ext}$$

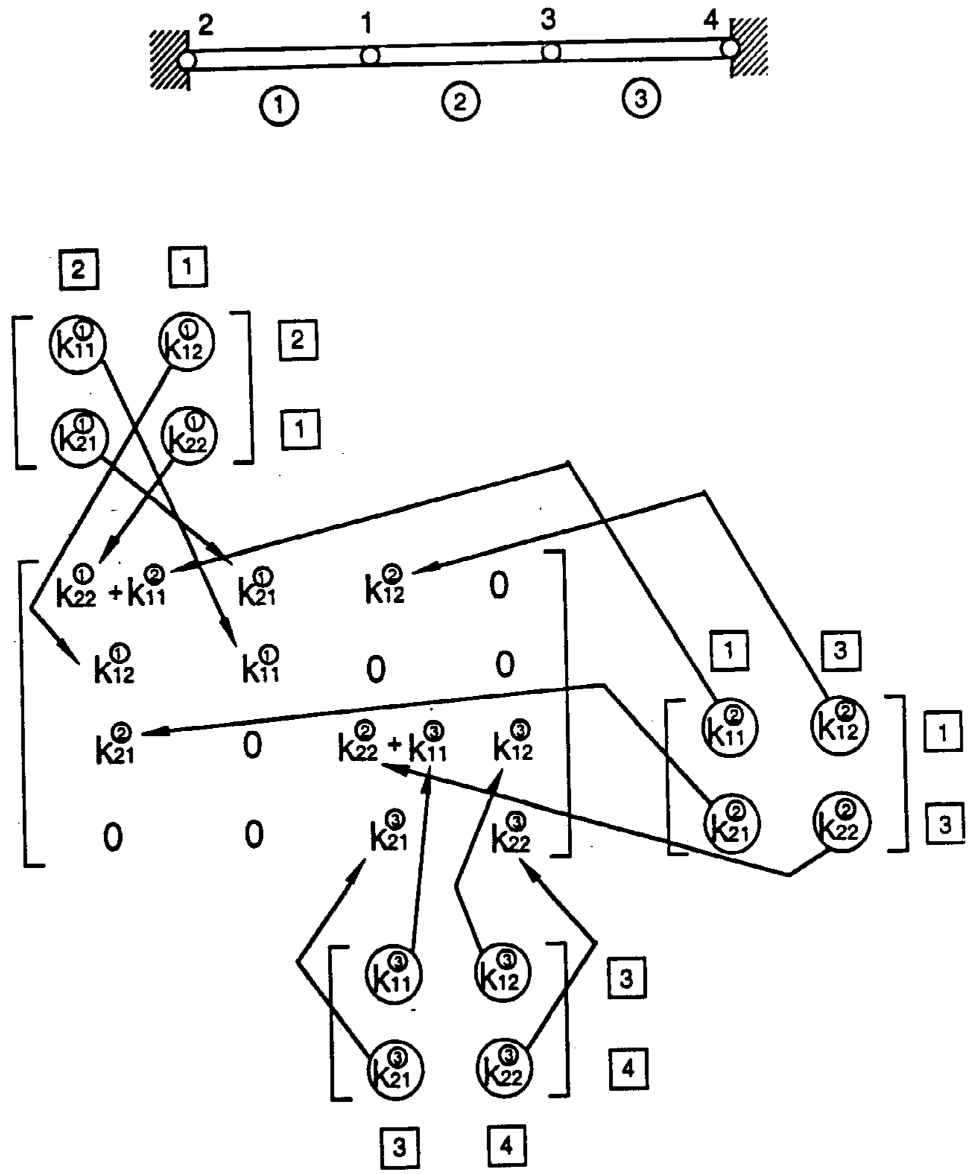
The augmented matrix technique was used here for the purpose of showing how the assembled equations are generated. It is important to remember that the stiffness equations are equilibrium equations and that they arise from summing the nodal forces of the elements. Compatibility is invoked when the nodal displacements of all elements are considered equal at the shared nodes.

Once these ideas are grasped, it is no longer necessary to assemble the equations by summing augmented matrices. If the contribution of all element stiffnesses are simply added into the total stiffness matrix according to their node numbers, or connectivity, we will obtain the total stiffness matrix. This is called direct assembly and can be easily visualized for sequential node numbering from Table 1D. An example of direct assembly for non-sequential node numbering is given in Table 2.

In Table 3, an example of direct assembly (sequential node numbering) and solution is given.

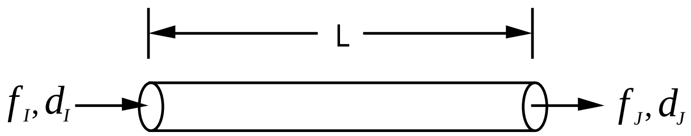
Remark: Note that, as mentioned previously, the assembly procedure requires that all nodal forces be consistently defined to be positive in the same direction.

TABLE 2 - DIRECT ASSEMBLY



Stiffness matrix of a rod element.

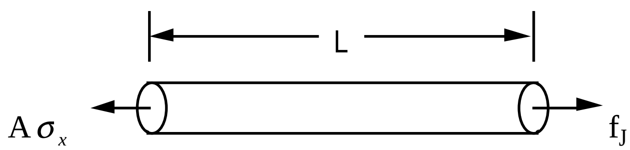
A rod element is a one dimensional element similar to a spring.



$$\mathbf{f_e} = \begin{Bmatrix} f_I \\ f_J \end{Bmatrix} \quad \mathbf{d_e} = \begin{Bmatrix} d_I \\ d_J \end{Bmatrix}$$

Cross-sectional area A, Young's modulus E

The element stiffness will now be derived by using equilibrium, the stress-strain law, and compatibility (strain-displacement equation).



By equilibrium of a free-body diagram of a section and the definition of stress

$$f_J = A\,\sigma \tag{1.12}$$

where it has been assumed that the stress is constant over the cross-section. Hooke's law gives

$$\sigma = E\,\varepsilon \tag{1.13}$$

The strain-displacement (or strain-elongation) equation gives

$$\varepsilon = \frac{\delta}{L} \tag{1.14}$$

The elongation can be expressed in terms of the displacements by

$$\delta = d_J - d_I \tag{1.15}$$

Substituting (1.13), (1.14) and (1.15) successively into (1.12) gives

$$f_J = A\,\sigma = A\,E\,\varepsilon$$

$$= \quad A\,E\,(\,d_J - d_I\,)\,/\,L \tag{1.16}$$

By equilibrium

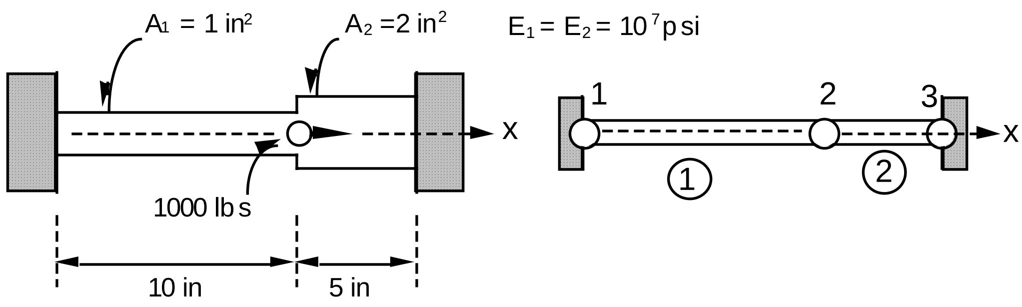
$$f_I \quad = \quad -f_J \quad = \quad A\,E\,(-d_J + d_I)\,/\,L \tag{1.17}$$

where the second equality follows from (1.16). Writing the above in matrix form gives

$$\begin{Bmatrix} f_I \\ f_J \end{Bmatrix} \quad = \quad \underbrace{\frac{AE}{L} \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix}}_{K_e} \begin{Bmatrix} d_I \\ d_J \end{Bmatrix} \tag{1.18}$$

The stiffness matrix of the rod is identical to that of the spring except that the spring constant is replaced by AE/L.

TABLE 3 - NUMERICAL EXAMPLE



PROBLEM DEFINITION

MESH

ELEMENT STIFFNESSES

$$A_1E_1/L_1 = 1.0 in^2 \times 10^7 \, psi/10.0 in = 10^6 \, lb/in$$

$$A_2E_2/L_2 = 2.0 in^2 \times 10^7 \, psi/5.0 in = 4 \times 10^6 \, lb/in$$

$$\mathbf{K}^{(1)} = 10^6 \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \qquad \mathbf{K}^{(2)} = 10^6 \begin{bmatrix} 4 & -4 \\ -4 & 4 \end{bmatrix}$$

ASSEMBLED STIFFNESS

$$\mathbf{K} = 10^6 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1+4 & -4 \\ 0 & -4 & 4 \end{bmatrix}$$

STIFFNESS EQUATIONS AND SOLUTION

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = 10^6 \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1+4 & -4 \\ 0 & -4 & 4 \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

Imposing the constraints that $d_1 = 0$ and $d_3 = 0$, yields one equation in one unknown, d_2

$$10^6 (lb/in)[5]\{d_2\} = \{1000\}(lbs) \text{ or } d_2 = 0.2\text{x}10^{-3}\text{in.}$$

Stiffness Method for 2D Trusses.

To develop a stiffness method for 2D trusses, we proceed as in 1D. We first develop an element stiffness which can be used for any element. For any particular mesh, a total stiffness matrix is assembled by direct assembly. The displacement constraints are then applied, which gives the governing linear algebraic equations which are solved for the displacements. Before developing the element stiffness, we need a few extra tools: transformations for the components of a vector and the rules for transformation of stiffness matrices.

Rotation Transformations. The objective of this section is to find expressions for components of vectors in different coordinate systems which are rotated relative to each other. The rotation in two dimensions is defined by the angle θ, which is positive counterclockwise from x to $\bar{x}$. Since the vector is the same physical quantity in any coordinate system, expressing it in either coordinate system is equivalent, i.e.

$$v_x\mathbf{i} + v_y\mathbf{j} = \bar{v}_x\bar{\mathbf{i}} + \bar{v}_y\bar{\mathbf{j}}$$

To obtain v_x in terms of $\bar{v}_x$, $\bar{v}_j$, take the scalar product of the above with $\mathbf{i}$. This yields

$$v_x(\mathbf{i}\cdot\mathbf{i}) + v_y(\mathbf{j}\cdot\mathbf{i}) = \bar{v}_x(\bar{\mathbf{i}}\cdot\mathbf{i}) + \bar{v}_y(\bar{\mathbf{j}}\cdot\mathbf{i})$$

Since $\mathbf{i}\cdot\mathbf{i}$=1 and $\mathbf{i}\cdot\mathbf{j}$=0, it follows from the above that

$$v_x = \bar{v}_x\,\bar{\mathbf{i}}\cdot\mathbf{i} + \bar{v}_y\,\bar{\mathbf{j}}\cdot\mathbf{i}$$

Similarly by taking the scalar product with $\mathbf{j}$ yields

$$v_y = \bar{v}_x\,\bar{\mathbf{i}}\cdot\mathbf{j} + \bar{v}_y\,\bar{\mathbf{j}}\cdot\mathbf{j}$$

For mnemonic and unifying purposes, the above two equations are written in the following matrix form

$$\begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = \begin{bmatrix} R_{\bar{x}x} & R_{\bar{y}x} \\ R_{\bar{x}y} & R_{\bar{y}y} \end{bmatrix} \begin{Bmatrix} \bar{v}_x \\ \bar{v}_y \end{Bmatrix} \tag{R.1}$$

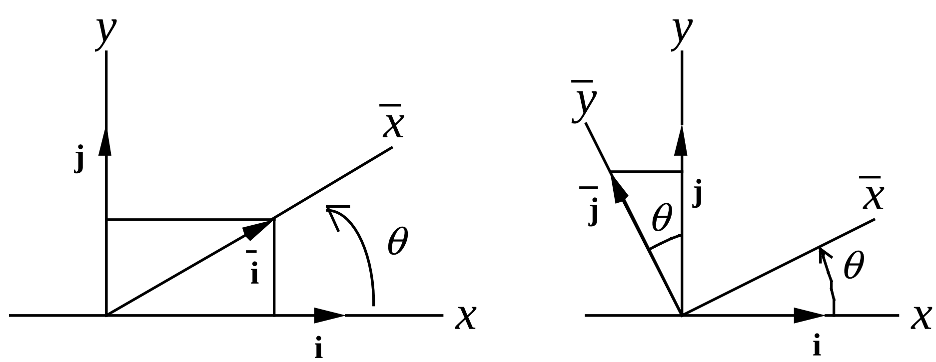
where

$$\begin{bmatrix} R_{\bar{x}x} & R_{\bar{y}x} \\ R_{\bar{x}y} & R_{\bar{y}y} \end{bmatrix} = \begin{bmatrix} \bar{\mathbf{i}}.\mathbf{i} & \bar{\mathbf{j}}.\mathbf{i} \\ \bar{\mathbf{i}}.\mathbf{j} & \bar{\mathbf{j}}.\mathbf{j} \end{bmatrix}$$

The subscripts of **R** have the following meanings: $R_{\bar{x}x}$ is the scalar product of $\bar{\mathbf{i}}$ and **i**, $R_{\bar{y}x}$ is the scalar product of $\bar{\mathbf{j}}$ and **i** etc. To remember this rotation transformation, note that the index of each term considered in a monomial appears on R, i.e. if an x component is being computed, then x appears as a subscript in each R term, and the other subscript is the same as the subscript of the v component which it multiplies:

$$v_x = R_{\bar{x}x}\,\bar{v}_{\bar{x}} + R_{\bar{y}x}\,\bar{v}_{\bar{y}}$$

The values of the terms R_{ij} can easily be ascertained from the figures shown below. Note the order of the subscripts does not matter since the scalar product is commutative $R_{\bar{x}x} = R_{x\bar{x}} = \bar{\mathbf{i}}^3\,\mathbf{i}$. The elements of the **R** matrix are often called direction cosines.



$$R_{\bar{x}x} = \bar{\mathbf{i}} \cdot \mathbf{i} = \cos\theta \qquad R_{\bar{y}x} = \bar{\mathbf{j}} \cdot \mathbf{i} = -\sin\theta$$
$$R_{\bar{x}y} = \bar{\mathbf{i}} \cdot \mathbf{j} = \sin\theta \qquad R_{\bar{y}y} = \bar{\mathbf{j}} \cdot \mathbf{j} = \cos\theta$$

In the above we have used the fact that **i**, **j** , $\bar{\mathbf{i}}$ and $\bar{\mathbf{j}}$ are unit vectors and the definition of the scalar product.

Inserting the values of **R** in terms of θ, we have

$$\mathbf{v} = \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{Bmatrix} \bar{v}_x \\ \bar{v}_y \end{Bmatrix} \qquad \text{(R.2)}$$

$c = \cos\theta \qquad s = \sin\theta$

The relationship between the components $\overline{\mathbf{v}}$ and $\mathbf{v}$ can be obtained by inverting **R,.** The inverse of **R** is given by

$$\mathbf{R}^{-1} = \frac{(\mathrm{cof}\,\mathbf{R})^{\mathrm{T}}}{\det(\mathbf{R})} = \frac{1}{c^2 + s^2}\begin{bmatrix} c & s \\ -s & c \end{bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix}$$

so

$$\begin{Bmatrix} \bar{v}_x \\ \bar{v}_y \end{Bmatrix} = \begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} \qquad \text{(R.3)}$$

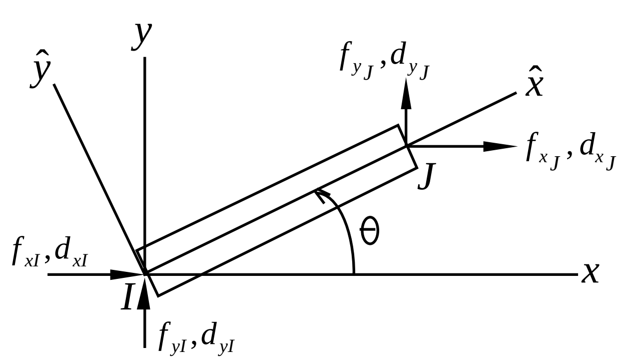
This relationship can also be developed directly from our rule on the transformation relationship

$$\begin{Bmatrix} \bar{x}_x \\ \bar{v}_y \end{Bmatrix} = \begin{bmatrix} R_{\bar{x}x} & R_{\bar{x}y} \\ R_{\bar{y}x} & R_{\bar{y}y} \end{bmatrix} \begin{Bmatrix} v_x \\ v_y \end{Bmatrix} \qquad \text{(R.4)}$$

Expressing the values of **R** in terms of θ and inserting the results in (R.4) yields the same expression as obtained by inverting **R**, (R.3).

Element Stiffness for a 2D Rod Structures made from of rod elements are often called trusses. Examples are of trusses are transmission towers and scaffolding used in construction and space structures.

The element stiffness matrix for a 2D rod element will now be develop. The bar (or rod) element in 2D is shown in the figure below. It can be seen that a local coordinate system has been constructed for the element so that the $\hat{x}$ coordinate lies along the axis of the



bar. The element stiffness matrix $\mathbf{K}_e$

in the global coordinate system will relate $\mathbf{f_e}$ to $\mathbf{d_e}$, where $\mathbf{f_e}^T = \{f_{xI}, f_{yI}, f_{xJ}, f_{yJ}\}$ and $\mathbf{d_e}^T = \{d_{xI}, d_{yI}, d_{xJ}, d_{yJ}\}$. This element stiffness matrix $\mathbf{K_e}$ will be a 4x4 matrix. The direct development of this stiffness matrix in terms of these components would be very difficult. Therefore, the element stiffness is first developed in the terms of the components in the local (element) coordinate system to relate

$$\widehat{\mathbf{f}}_e = \widehat{\mathbf{K}}_e \, \widehat{\mathbf{d}}_e \tag{R.9a}$$

Because the truss element is a two-force body, the internal forces are directed along the axis of the element (i.e., the $\hat{x}$ axis) and it follows that, in the local coordinate system, $\hat{f}_{yI} = \hat{f}_{yJ} = 0$. Furthermore, only the axial elongation ($\hat{d}_{xJ} - \hat{d}_{xI}$) gives rise to internal forces. In the element coordinate system, then, the element stiffness matrix is identical to that of (1.18) except that we represent the internal forces in a two-dimensional coordinate system, i.e., with zeroes added to reflect these conditions on the internal forces, which gives

$$\begin{Bmatrix} \hat{f}_{xI} \\ \hat{f}_{yI} \\ \hat{f}_{xJ} \\ \hat{f}_{yJ} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} \hat{d}_{xI} \\ \hat{d}_{yI} \\ \hat{d}_{xJ} \\ \hat{d}_{yJ} \end{Bmatrix} \tag{R.9b}$$

To develop the stiffness matrix in terms of global components, a relationship between local and global components must be determined. To develop these relationships, the transformation laws for vectors is used. Using the transformation equations, (R.4), at each of the nodes gives

$$\begin{aligned} \hat{d}_{yI} &= -sd_{xI} + cd_{yI} \qquad & \hat{d}_{xI} &= cd_{xI} + sd_{yI} \\ \hat{d}_{yJ} &= -sd_{xJ} + cd_{yJ} \qquad & \hat{d}_{xJ} &= cd_{xJ} + sd_{yI} \end{aligned}$$

where $c = \cos\theta$, $s = \sin\theta$. The matrix of nodal displacements in terms of the global components is written in the form $\mathbf{d_e}^T = \{d_{xI}, d_{yI}, d_{xJ}, d_{yJ}\}$. Writing the above expressions in the matrix form gives

$$\begin{Bmatrix} \hat{d}_{xI} \\ \hat{d}_{yI} \\ \hat{d}_{xJ} \\ \hat{d}_{yJ} \end{Bmatrix} = \underbrace{\begin{bmatrix} c & s & 0 & 0 \\ -s & c & 0 & 0 \\ 0 & 0 & c & s \\ 0 & 0 & -s & c \end{bmatrix}}_{\Downarrow \atop T} \begin{Bmatrix} d_{xI} \\ d_{yI} \\ d_{xJ} \\ d_{yJ} \end{Bmatrix} \tag{R.10}$$

The third and fourth entries of rows 1 and 2 of the **T** matrix are zero because the transformation relations at node I do not involve the nodal displacements at node J: similarly, the first and second entries of rows 3 and 4 are zero.

The stiffness transformation rule is now developed. Note that the same relationship as (R.10) applies to the force components, i.e.

$$\hat{\mathbf{f}}_{\mathbf{e}} = \mathbf{T}\mathbf{f}_{\mathbf{e}}$$

Since **T** is an orthogonal matrix, from EA1 you know that $\mathbf{T}^{-1} = \mathbf{T}^{\mathbf{T}}$ and we can write

$$\mathbf{f}_{\mathbf{e}} = \mathbf{T}^{-1}\hat{\mathbf{f}}_{\mathbf{e}} = \mathbf{T}^{\mathbf{T}}\hat{\mathbf{f}}_{\mathbf{e}}$$

Then using (R.9a) and (R.10), we have

$$\mathbf{f}_{\mathbf{e}} = \mathbf{T}^{\mathbf{T}}\hat{\mathbf{f}}_{\mathbf{e}} = \mathbf{T}^{\mathbf{T}}\hat{\mathbf{K}}_{\mathbf{e}}\hat{\mathbf{d}}_{\mathbf{e}} = \mathbf{T}^{\mathbf{T}}\hat{\mathbf{K}}\mathbf{T}\mathbf{d}_{\mathbf{e}}$$

Thus the element stiffness in the global coordinate system is given by:

$$\mathbf{K}_{\mathbf{e}} = \mathbf{T}^{\mathrm{T}}\widehat{\mathbf{K}}_{\mathbf{e}}\mathbf{T}$$

$$= \frac{AE}{L}\begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \tag{R.11}$$

The relationship between element nodal forces and nodal displacements can then be written using the matrix defined in (R.11) in the form

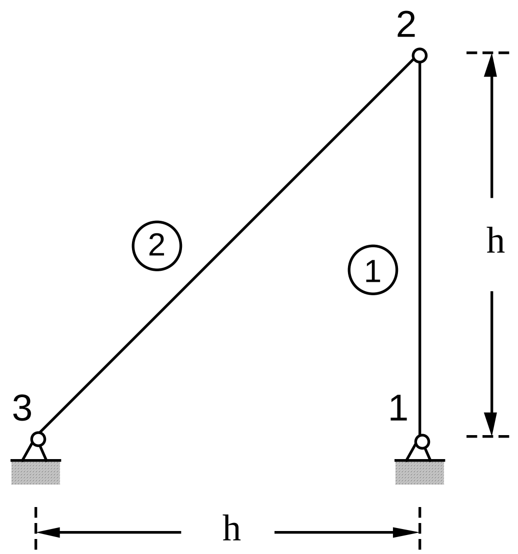
$$
\begin{Bmatrix} f_{xI} \\ f_{yI} \\ f_{xJ} \\ f_{yJ} \end{Bmatrix} = \frac{AE}{L} \begin{bmatrix} c^2 & cs & -c^2 & -cs \\ cs & s^2 & -cs & -s^2 \\ -c^2 & -cs & c^2 & cs \\ -cs & -s^2 & cs & s^2 \end{bmatrix} \begin{Bmatrix} d_{xI} \\ d_{yI} \\ d_{xJ} \\ d_{yJ} \end{Bmatrix} \tag{R.12}
$$

The element stiffness matrix is symmetric. However, it is symmetric only when the terms in the nodal force and nodal displacement matrix have been ordered so that their scalar product yields work. It can easily be seen that if the first and second terms in $\mathbf{d}_e$ are interchanged and if the terms of $\mathbf{f}_e$ are not rearranged, then the first and second columns of $\mathbf{K}_e$ would be interchanged and the resulting element stiffness would no longer be symmetric.

Assembly in 2D and 3D.

Assembly in multi-dimensional problems is carried out as in one dimensional problems: the contributions of all element stiffnesses are added into the appropriate locations of the total stiffness matrix according to the node numbers of the element. However, to take advantage of the fact that there are more than one degree-of-freedom per node, the element and global stiffnesses can be partitioned into submatrices which give all the stiffness terms pertaining to the interactions of two nodes. Thus the stiffness matrices are partitioned into 2x2 submatrices in 2D, 3x3 submatrices in 3D. The partitioning for a 2D example is shown in Table 4, which gives the stiffness assembly and the formulation of the stiffness equations.

TABLE 4 – 2 DIMENSIONAL EXAMPLE



A, E constant

AE = 10^6

h = 10

no units

$\theta_1 = 90^\circ$ (could be -90°), $\theta_2 = 45^\circ$

	I	J	θ	c	s
1	1	2	90°	0	1
2	3	2	45°	$\sqrt{2}/2$	$\sqrt{2}/2$

$$\mathbf{K}^{(1)} = \frac{AE}{h}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix} \qquad \mathbf{K}^{(2)} = \frac{AE}{2\sqrt{2}h}\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 1 & 1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

ASSEMBLED STIFFNESS

$$\mathbf{K}=\frac{AE}{h}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & \sqrt{2}/4 & \sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 \\ 0 & -1 & \sqrt{2}/4 & 1+\sqrt{2}/4 & -\sqrt{2}/4 & -\sqrt{2}/4 \\ 0 & 0 & -\sqrt{2}/4 & -\sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 \\ 0 & 0 & -\sqrt{2}/4 & -\sqrt{2}/4 & \sqrt{2}/4 & \sqrt{2}/4 \end{bmatrix}$$

EQUILIBRIUM EQUATIONS WITH CONSTRAINTS IMPOSED

$$\frac{AE}{h}\begin{bmatrix} \sqrt{2}/4 & \sqrt{2}/4 \\ \sqrt{2}/4 & 1+\sqrt{2}/4 \end{bmatrix}\begin{Bmatrix} d_{x2} \\ d_{y2} \end{Bmatrix}=\begin{Bmatrix} f_{x2}^{ext} \\ f_{y2}^{ext} \end{Bmatrix}=\begin{Bmatrix} 10 \\ 0 \end{Bmatrix}$$

SOLUTION

$$\begin{Bmatrix} d_{x2} \\ d_{y2} \end{Bmatrix}=\frac{h}{AE}\begin{Bmatrix} 10+20\sqrt{2} \\ -10 \end{Bmatrix}$$

STRESSES

$$\sigma=\frac{\overline{f}_{xj}}{A}=\frac{E}{h}\begin{bmatrix} -c & -s & c & s \end{bmatrix}\{d\}_e$$

ELEMENT 1 - STRESS COMPUTATION

$$\sigma=\frac{E}{h}\begin{bmatrix} 0 & -1 & 0 & +1 \end{bmatrix}\frac{h}{AE}\begin{Bmatrix} 0 \\ 0 \\ 20\sqrt{2}+10 \\ -10 \end{Bmatrix}=\frac{-10}{A} \qquad \text{(compressive)}$$

ELEMENT 2 - STRESS COMPUTATION

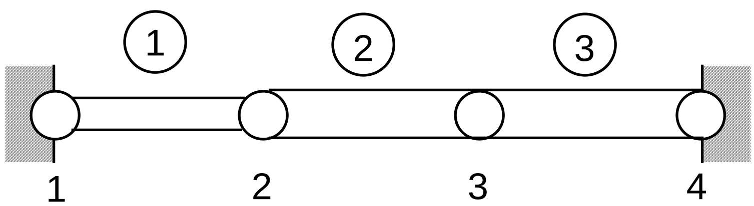
$$\sigma = \frac{E}{h}\left[-\frac{\sqrt{2}}{2} \quad -\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2}\right]\frac{h}{AE}\begin{Bmatrix} 0 \\ 0 \\ 20\sqrt{2}+10 \\ -10 \end{Bmatrix} = \frac{10\sqrt{2}}{A} \qquad \text{(tensile)}$$

MATLAB for Finite Element Analysis

To illustrate the application of MATLAB to finite element analysis, consider the program for the analysis of one dimensional rods given at the end of this Section. This program can treat any one-dimensional rod with any combination of cross-sectional areas, Young's moduli and loads once the data are specified. The same methods as described in class are used, so the program calculates element stiffnesses, assembles the total stiffness and the solves the equations.

The first set of statements gives the finite element data. The data has been set up so that the element and nodal data are generated automatically for **numele** sequential elements with **numnod=numele+1** sequential , equally-spaced nodes between x-coordinates 0 and 1. The coordinates of the nodes are stored in the array **x**. The colon is used to generate these nodal coordinates.

The array **node** stores the connectivity of the node numbers of the elements. The colon is used to assign the first node number of the element a sequential value from 1 to **numele**, the second node number from 2 to **numele+1**. For the case when **numele=3,** this corresponds to the following mesh:



The following describes the other data:

area= cross-sectional areas of the elements, area(e) is te area of element e;
young=Young's modulus; young(e) is the moduls for element e;
ifix= an array which describes to the program whether a node is fixed or free;
force=the applied forces; force(i) is the force applied to node i.

You can make your own data, changing the number of elements and material properties to solve other problems. The solution procedure is independent of the ordering of the node numbers. With a little thought, you can probably think of several ways in which ths code could be made more general (good for solving problems but not necessarily for illustrating the procedure initially!).

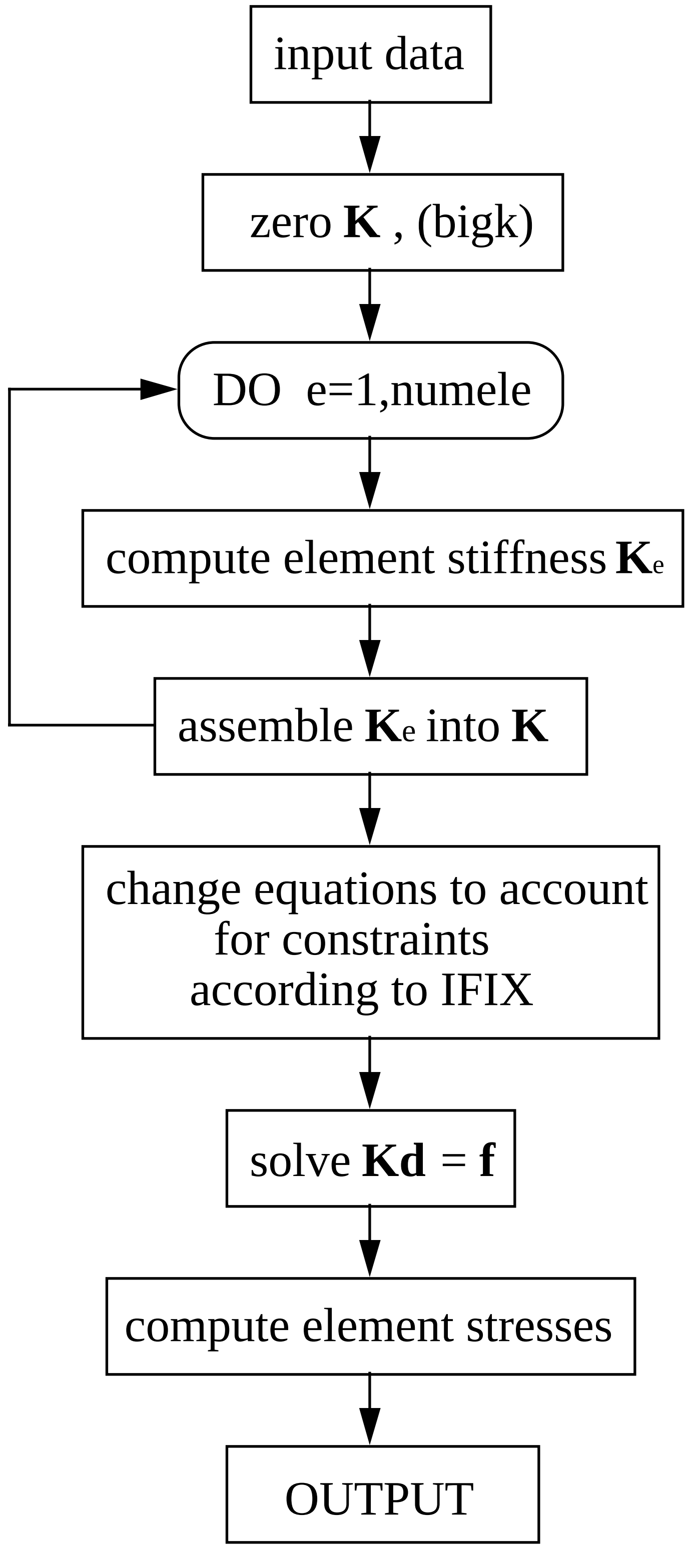
Sample MATLAB Code for 1D rod problems

```
% One dimensional finite element program
% INPUT DATA
% this data automatically describes a line of 2-node elements
% with support conditions at both ends and a uniform load; the
% number of elements can be varied by changing numele
numele=6, numnod=numele+1
% x-coordinates of nodes
x= 0:1/numele:1
% node stores the nodes of all elements
node= [1:numele; 2:numele+1]
area=[2.0*ones(1,numele)]
young=[1.E7*ones(1,numele)]
% support conditions, ifix(i)=1 if node i is fixed, else zero
ifix=[1,zeros(1,numele)]
ifix(numnod)=1
% applied forces
force=[1000/numele*ones(1,numnod)]
%
% zero bigk matrix to prepare for assembly
bigk=[zeros(numnod,numnod)]
%
%  loop over elements
%
for e=1:numele
% compute element length
        length=x(node(2,e))-x(node(1,e))
        c=young(e)*area(e)/length
% compute element stiffness
        ke=[c,-c;-c,c]
%
% assemble ke into bigk

bigk(node(1,e),node(1,e))=bigk(node(1,e),node(1,e))+ke(1,1);
bigk(node(1,e),node(2,e))=bigk(node(1,e),node(2,e))+ke(1,2);
bigk(node(2,e),node(1,e))=bigk(node(2,e),node(1,e))+ke(2,1);
bigk(node(2,e),node(2,e))=bigk(node(2,e),node(2,e))+ke(2,2);
end
```

```
% support conditions (boundary conditions)
for n=1:numnod
   if (ifix(n) == 1)
        bigk(n,n)=1E+30
        force(n)=0
   end
end
%
% solve stiffness equations
disp=force/bigk
%
%  plot displacements
subplot(211), plot(x,disp)
% compute stresses
for e=1:numele
% compute element length
        length=x(node(2,e))-x(node(1,e))
  elong=disp(node(2,e))-disp(node(1,e))
  stress(2*e-1)=young(e)*elong/length
  stress(2*e)=stress(2*e-1)
  xx(2*e-1)=x(node(1,e))
  xx(2*e)=x(node(2,e))
end
subplot(212), plot(xx,stress)
```

FLOWCHART OF FEM PROGRAM



Graphics Window

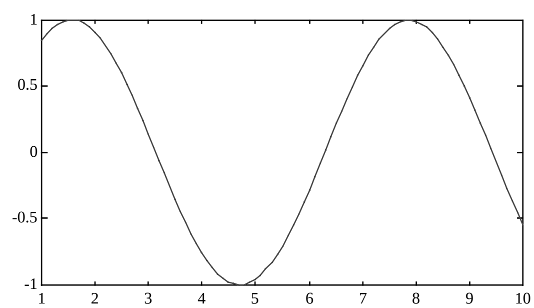
The following note on the use of the graphics window in MATLAB may be useful. Before you can use the graphics window, you have to have something to graph, so in the command window you might do the following:

»x=[1:.1:10];	a vector of input values
»y=sin(x);	sin(vector) computes the sine of the vector component by component

Then, to plot:

»plot(x,y)	

And in the graphics window, you will see:



Labels and such can be added by using the pull-down menu "Graph" or as special strings inside the "plot" command (see "help" for details). If you print the graph directly by using "Print" from the "File" menu, it will look better than this, because Matlab will do some interpolation and smoothing.

To open up a new figure window and plot another graph, while still retaining the first plot, type the following commands:

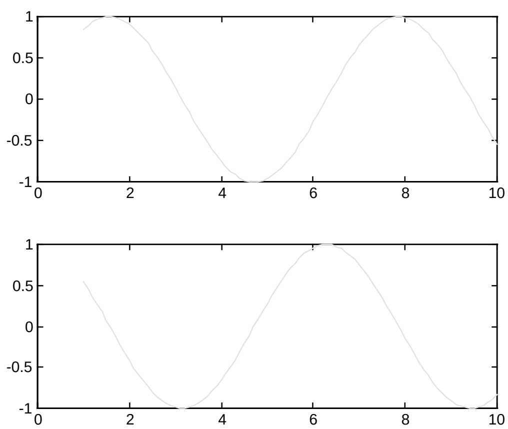
»z=cos(x);	computes the sine of the vector component by component
»figure	opens new figure window
»plot(x,z)	plots cos(x) vs x in new figure window

The new graphics window will show the new plot; however, the original sin(x) plot is still present in the earlier figure window – it may be hidden immediately behind the cos(x) window. To move the cos(x) window to see the sin(x) window, click and hold with the mouse on the topmost bar and drag the cos(x) window to another location. The sin(x) window should be revealed. Note that if you had typed the "plot(x,z)" command without typing the "figure" command first, the cos(x) plot would replace the sin(x) plot in the first figure window.

To plot multiple plots in the same figure window, side by side or in a column, use the "subplot" command:

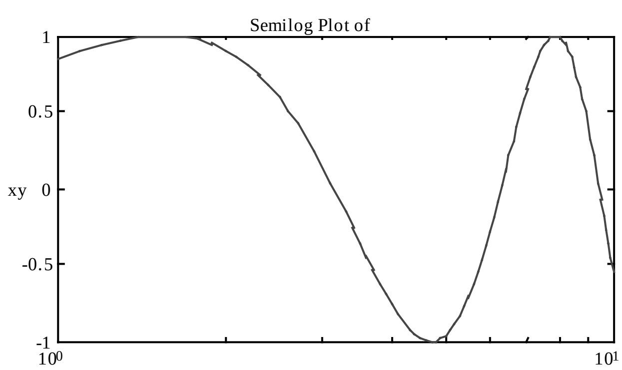
»subplot(2,1,1), plot(x,y)	plots sin(x) in row 1, column 1 of a 2x1 plotting grid
»subplot(2,1,2), plot(x,z)	plots cos(x) in row 2, column 1 of a 2x1 plotting grid

Note that the new plots have gone into the figure window that had contained the cos(x) curve orginally, since we had not used the "figure" command before typing the subplot commands. Your graphics window should show:



Use the "help subplot" command to see the full array of possiblities and guidelines for multiple axis plotting. There are lots of other things you can do with Matlab's graphics routines, including changing line types, axis limits, etc. You also have several types of "graph paper" available, including semilog and loglog, which can be useful in plotting frequency responses. An example of using semilog paper would be:

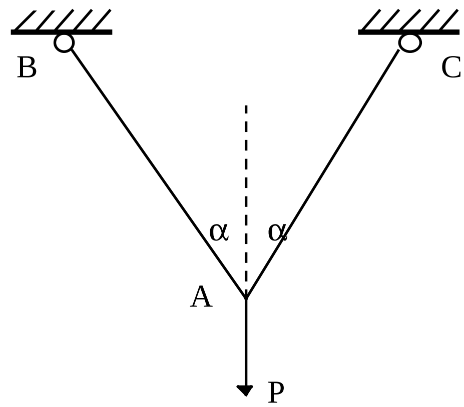
»semilogx(x,y)	



APPENDIX

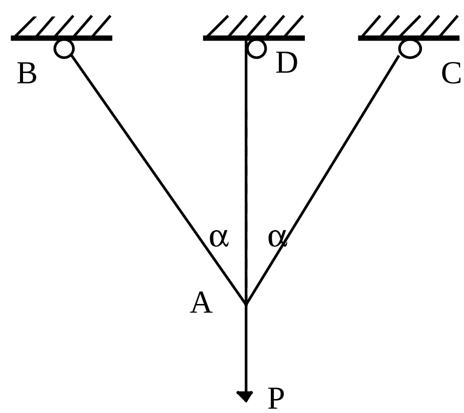
MECHANICS OF DEFORMABLE BODIES

Consider the two trusses shown in the figures:



Case (a)

Case (b)

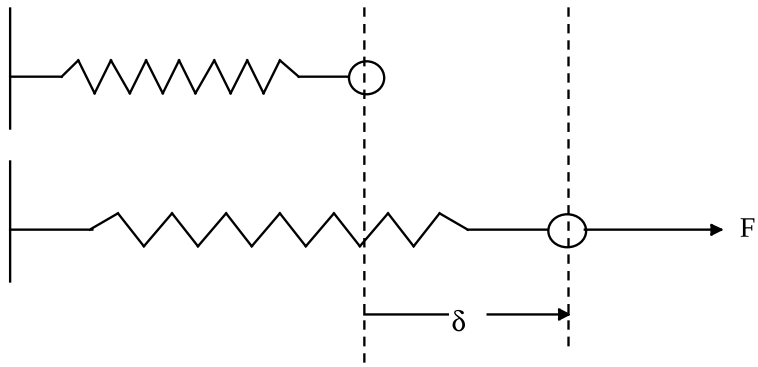


We can solve for the forces carried by the truss members in case (a) from conditions for static equilibrium - all you have to do is consider the equilibrium of the hook at A. There are two unknown forces in the rods AB and AC which can be solved for in terms of the applied load P and the angle α. You should do this on your own right now.

Static equilibrium considerations are insufficient to obtain the forces carried by truss (b)! This is because we now have unknown forces in the three rods AB, AC and AD, but we still have only two useful equations from static equilibrium (the moment equilibrium equation is trivially satisfied.) Problems such as this are called *statically indeterminate*. It turns out that the only way we can get the forces in the members of truss (b) is if we look into the deformation in the members of the truss due to the applied loads. That is, we can no longer afford to neglect the deformation of structures—as we have been doing thus far by assuming that structures or bodies were rigid. Of course, in some instances, the deformation of an object might be of inherent interest to us; for example, the deformation of a shock absorber or a bumper in a car, or the deformation (stretch) of a bungee cord etc. So we are now going to relax our assumption of rigid bodies, and inquire into the deformation of real bodies under the action of applied forces.

A.1 Springs:

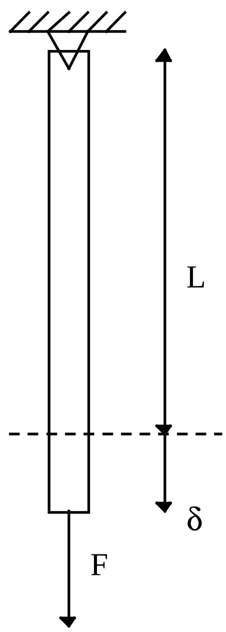
We have actually already encountered one kind of a deformable body in our adventures so far —a linear spring which exerts a force that is proportional to its stretch from its unstretched state:



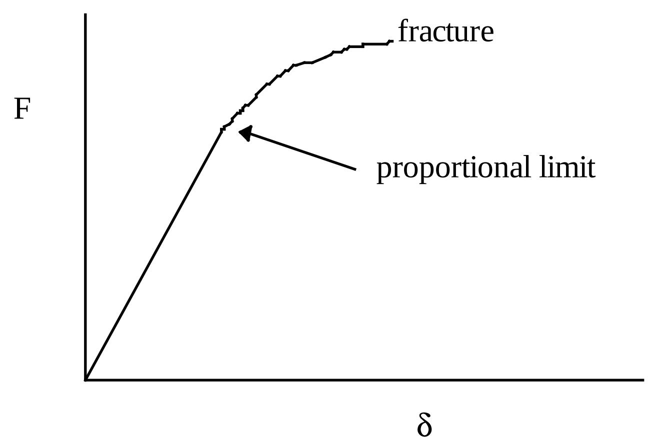
$$F = k\,\delta$$

where k is called the spring constant of the spring.

A.2 Axial Stretching of Rods:



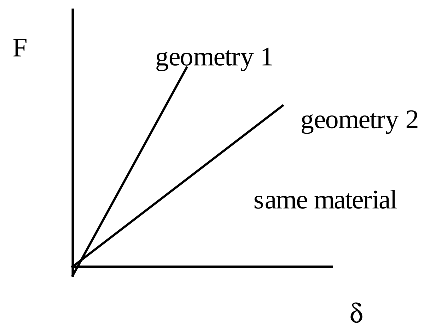
Consider a straight rod of length L and uniform cross-sectional area A made of some material. Let one end of the rod be pinned to a wall, and the other end be subjected to an applied load F. Because of the applied load F, the rod is found to stretch by an amount δ. If we were to increase the load F, the stretch is found to increase as well. In fact, if we plot the stretch of the rod against the force applied, we find a plot that looks something like this:

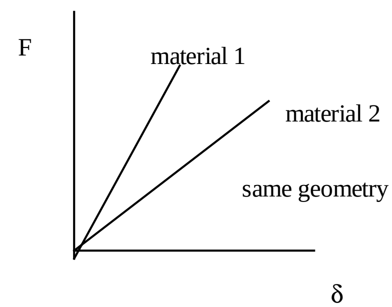


This experimental fact was discovered by Robert Hooke in 1678. Not being quite sure that he was onto something good, but still wanting to establish priority, he stated his discovery— "Hooke's law"— in the form of an anagram: ceiiinosssttuv. This was not of much help to others who did not know what he was talking about. When someone unscrambled the anagram it turned out to be: ut tensio sic vis which loosely translates to: the extension is proportional to the force. That is, an axial rod behaves like a linear spring. It turns out that what Hooke found is quite true but we need to fix it a little bit for it to be of use to us.

Problems with Hooke's law as stated by Hooke:

- True only for a certain class of materials: for instance, it is true for most engineering materials (steel, aluminum etc), but is not really accurate for rods made out of animal tissue for instance. Materials for which Hooke's law holds are called **linear elastic materials**.
- True only upto a point: beyond a certain amount of elongation the force is no longer proportional to the elongation, but may be non-linearly related, and at some point the rod will break (fracture).





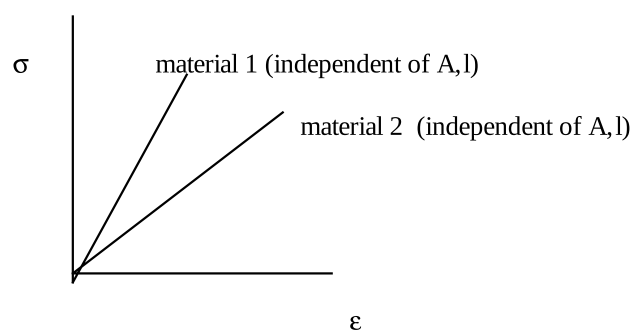
- Cannot distinguish between material and geometric effects: That is if we did two sets of experiments where in one set we test several rods of the same material but different geometry (cross sectional area A and length L), and in the other we test several rods of the same geometry but made from different materials, we get different proportionality constants for the force-stretch relation. So we recognize that the force is proportional to the elongation but the proportionality constant depends both on the material and the geometry.

In order to separate the effect of geometry and material, we scale out the geometric effect by defining:

Stress: $\sigma = F/A$ = (force) / (cross-sectional area) $[N.m^{-2}]$

Strain: $\varepsilon = \delta/L$ = (elongation) / (original length) [dimensionless]

Then, if we were to plot stress vs strain, we find:



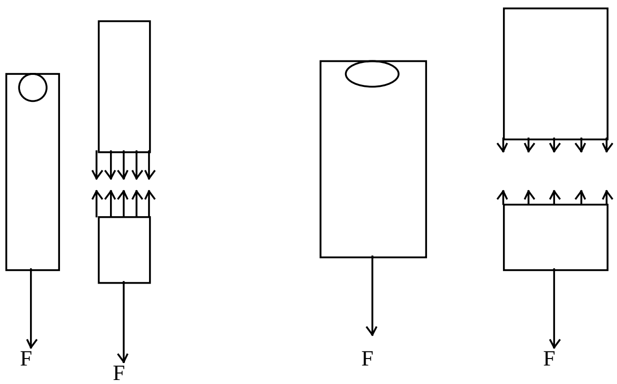
That is, the stress is proportional to the strain in axial stretching of a rod, and the proportionality constant is just a material property.

Amended version of Hooke's law: Stress is proportional to strain for a certain class of materials and for small deformations:

$$\sigma = E\,\varepsilon$$

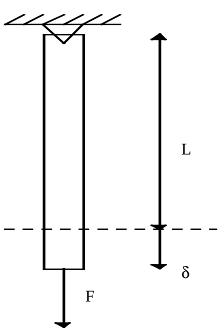
where E is called the Young's modulus [$N.m^{-2}$], and it is a property of the material. It is a measure of the stiffness of the material and has different values for different materials.

Remark: Both stress and strain have a deeper meaning, and have significant character! Stress turns out to be a useful measure of the *intensity* with which the atoms or molecules of a material resist the applied load. The reason that one part of the rod does not break away (hopefully) from the other is because of these resisting forces that arise from atomic/molecular forces.



Consider two rods A and B made of the same material but A is fatter than B. Since, a fatter rod has more atoms/molecules across which to spread out the force that is needed to resist the applied load, A is stressed less, and can actually carry a higher applied force. However, the maximum *stress* at which a fat rod and a thin rod made of the same material will break—or go non-linear or "plastic" will be the same and depends only on the material. We will, however, not go into that right now.

Case 1: Uniform Rod in Tension



- uniform cross-sectional area A, original length L
- Neglect weight of rod itself
- Young's modulus E

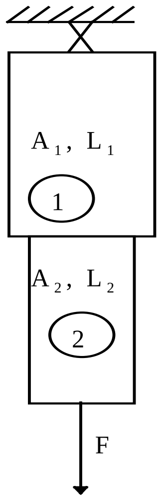
Since the stress: $\sigma = E\,\varepsilon \rightarrow P/A = E\,\delta/L$
we have:

$$F = [AE/L]\ \delta \qquad \text{or} \qquad \delta = FL/AE$$

where AE/L is called the axial stiffness (and is the equivalent of the spring constant for a linear spring).

Remark: If the force F were to act into the rod, it is called compressive, and the rod shrinks in length or compresses. Compression can be thought of as opposite (negative) of tension. For most materials, the amount of rod compression is proportional to the applied compressive force, and the proportionality constant is the same as that of tension. Therefore, the above axial-force vs elongation relation can be used in both compression and tension, where we treat negative forces and negative stretches as meaning compression..

Case 2: 2-Stepped Rods:



- Rod of two segments of length L_1 and L_2 and cross-sectional area A_1 and A_2 made of material whose Young's modulus is E_1 and E_2 respectively.
- Neglect weight of the rod itself. Determine total stretch of the rod due to the applied load F.

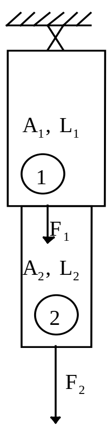
By making a cut in each segment, we note that the force in each segment in this case is F. But the stresses and elongations are different.

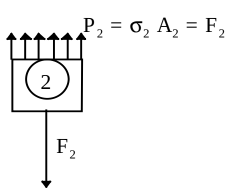
stretch of segment 1 is $\delta_1 = FL_1 / A_1E_1$
stretch of segment 2 is $\delta_2 = FL_2 / A_2E_2$

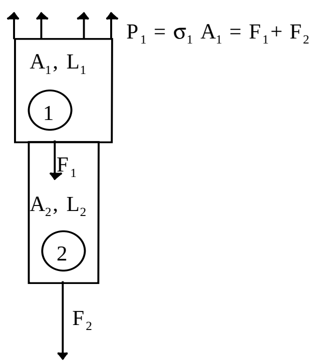
Total stretch of the stepped rod is: $\delta = \delta_1 + \delta_2 = FL_1 / A_1E_1 + FL_2 / A_2E_2$

Remark: In this case, the forces on each segment were the same. But this need not be the case if an additional force were to act at, say, just under segment 1. In this case,

$$\delta = \delta_1 + \delta_2 = P_1L_1 / A_1E_1 + P_2L_2 / A_2E_2$$







Case 3: N-Stepped Rods:

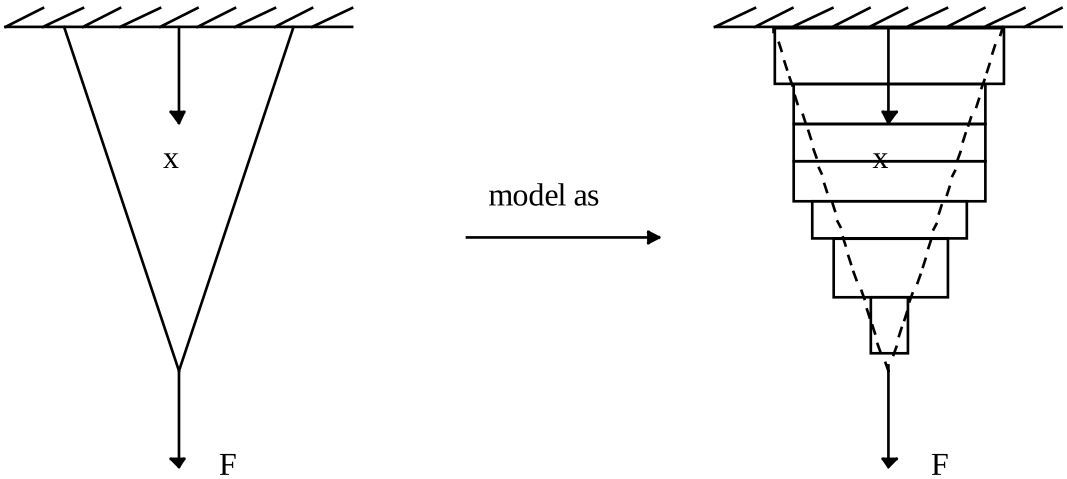
We can easily extend the above to a rod with N-steps.

$$\delta = \sum_{i=1}^{N} \frac{P_i L_i}{A_i E_i}$$

where P_i is the net axial force (also called the net internal force) on the ith segment whose length is L_i, cross-sectional area is A_i, and is made of a material whose Young's modulus is E_i. The idea is that we treat each segment of the rod as a uniform rod over which the net cross-sectional force as well as the area are constant.

Case 4: Rods with Continuously Varying Cross-Section and / or Loading:

Next, let us consider the case of a rod with varying cross-section such as the conical rod shown. By making cuts at several locations along the stalactite, convince yourself that in this case, even though the net cross sectional force is the same everywhere, the stress at each cross-section of the rod is not the same. Or else consider a rod with uniform cross-section but whose weight is not negligible. In this case, the net cross-sectional force is different at different locations.



It is possible for us to approximate the continuously varying rod as being made up of N segments each of which is uniform and is of length Δx. The stretch of any segment is just:

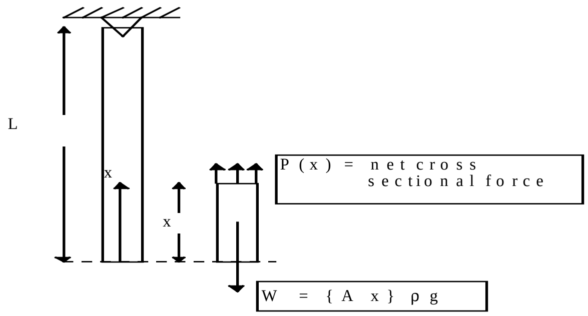
$$\Delta\delta = \frac{P}{EA}\ \Delta x\,.$$

Then the total stretch of the rod is just the sum of the stretches of each segment. The approximation becomes exact as we shrink the segment lengths Δx --> 0. In this limit, total stretch is just

$$\delta \ = \sum_{all\ \ segments}\Delta\delta \ = \sum_{all\ \ segments}\frac{P}{EA}\ \Delta x \xrightarrow[as\ \ \Delta x\to 0]{} \int_0^L \frac{P}{EA}\ dx$$

{Recall the meaning of an integral as a sum}

Example 1: Stretching of a Uniform Rod Under its Own Weight:



- Rod with uniform cross sectional area A and length L
- Density of material is ρ.
- Young's modulus of material is E.
- The only load is due to its own weight.

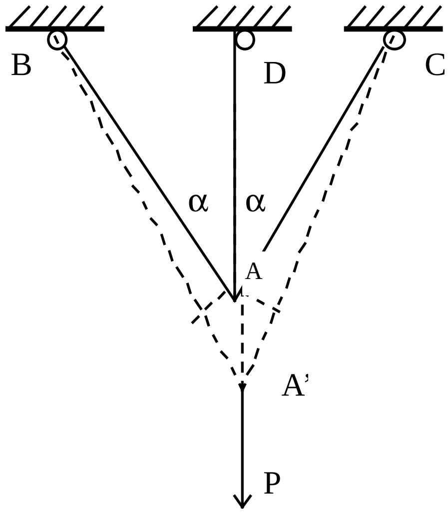
Locate the origin of the x axis at the bottom as shown.

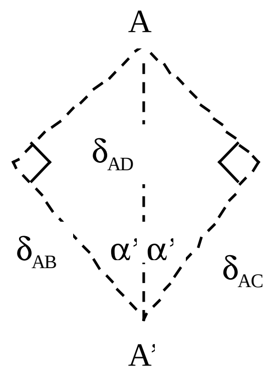
To get the net cross-sectional force at any cross-section, imagine making a "cut" at a distance x from the bottom and look at the part below the cut. The net cross-sectional force P(x) due to the internal forces must balance the weight of the chunk of material below it. So P(x) = ρgAx. The total stretch of the rod under its own weight is therefore:

$$\delta = \int_o^L \frac{P(x)}{EA}dx = \int_o^L \frac{\rho g A x}{EA}dx = \frac{\rho g L^2}{2E}.$$

Example 2: Now let us return to the truss problem that we could not solve earlier because it was statically indeterminate.

Given that all three rods are linear elastic of Young's modulus E, and cross-sectional area A, and that the rods AB and AC are of length L, and rod AD is vertical, determine the forces carried by the three rods due to the applied load P at A. (Neglect the weight of the rods).





From the equilibrium of the pin at A, we get

$$\sum F_x = 0 \Rightarrow P_1 = P_2$$

:

$$\sum F_y = 0 \Rightarrow P_1 \cos\alpha + P_2 \cos\alpha = P$$

which are two equations for three unknowns (insufficient). Problem is statically indeterminate. Need to look for an additional condition. Requiring that the three rods not break apart, we find that the stretches of the rods are not independent but are related.

We recognize that from the symmetry of the problem (rods AB and AC are identical), rod AD stretches by δ_{AD} such that it is still vertical. Rods AB and AC then must also stretch appropriately.

If we assume that the deformations are small compared to the lengths of the rods, then, we find from the figure (shown grossly exaggerated) that the stretches are approximately related through:

$$\delta_{AB} = \delta_{AC} = \delta_{AD} \cos\alpha' \approx \delta_{AD} \cos\alpha$$

where again the assumption of small deformation allows us to say that the angle α' is approximately the same as α.

Recasting this in terms of the forces, we have:

$$\frac{P_1 L}{AE} = \frac{P_2 L}{AE} = \frac{P_3 (L\cos\alpha)}{AE} \cos\alpha$$

which gives us the additional restriction needed to solve for the forces in the rods.

$$P_1 = P_2 = \frac{P\cos^2\alpha}{1+2\cos^3\alpha}; \qquad P_3 = \frac{P}{1+2\cos^3\alpha}.$$

Remark: If you study what we have just done carefully, you will notice that there are three things needed to solving these problems. We impose
(a) equilibrium
(b) compatibility—or the geometric constraints
(c) the material response (Hooke's law)

In fact, to determine the forces (stresses) and stretches (displacements) of all structures, we follow exactly the same procedure. An N-stepped rod or a system of N linear springs connected along a line can be analyzed by the process above. Trusses, which are made of rods but now in two or three-dimensions can also be analyzed in exactly the same way, except that now the compatibility (geometry) part can become somewhat complicated.
It turns out that we can automate this process rather easily, and this is a great convenience when we are dealing with large structures. This leads us to the topic of matrix analysis of structures which forms part of what is called the finite element method.