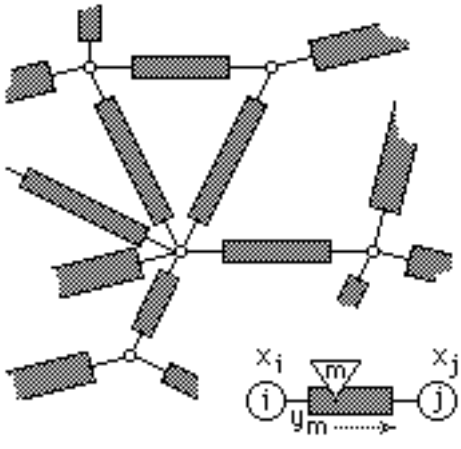
FINITE ELEMENTS

The vast majority of shapes which occur in Engineering - whether they be chunks of stressed metal or volumes of flowing gases - are complex three-dimensional continua which cannot be represented adequately by the simple closed-form mathematical models so beloved of engineering students. These theoretical models also cannot portray the appreciable non-linear or anisotropic material characteristics which are often met with in practice. For example, although complex mathematical transformations enable the flow of ideal fluids around certain aerofoils to be analysed, the very real effects of viscosity may render such analyses impracticably inaccurate.

In such situations therefore, we must resort to numerical methods in which the prototypical continuum of infinitesimal material particles is represented by an approximately equivalent assembly of inter-connected discrete elements which are each so simple that they can be treated individually as mathematical continua. There are a number of methods whereby such networks can be analysed - numerical solution of differential equations, finite differences, finite elements, boundary elements, relaxation techniques, and so on. We have chosen to demonstrate the Finite Element Method (FEM) as a typical powerful approach which can handle equilibrium, eigenvalue and propagation problems - though we shall restrict our considerations to equilibrium applications in linear elasticity.

Finite Elements involve a particular type of network. A network is an arrangement of 'elements' interconnected between 'nodes', such as an electrical system in which the elements are resistors, batteries, etc, or a fluid network where the elements may be pumps or fans, pipes, valves or mine airway resistances, and so on. Networks abound also in non-engineering disciplines - in economics, in political and the social sciences, and in decision theory to name but a few. This ubiquity is sufficient reason to first examine networks in their own right to see how best to analyse them - this examination will lead naturally into the concept of Finite Elements.

Linear 1-Networks



The sketch illustrates portion of a net which consists of two-noded elements, 'M' in number and labelled 1, 2 . . m . . . M, arranged between 'N' nodes which are indexed 1, 2 . . i, j . . . N. The typical m'th element, lying between the i'th and j'th nodes, is also shown. The net variables are :-

'element' or *'through'* variables, 'y' say, exemplified by the current or flow, y_m, through the m'th element, and

'node' or *'across'* variables, 'x' say, typified by the voltages or heads, x_i and x_j, at the i'th and j'th nodes respectively.

Analysis of the net as a whole requires determination of the complete vectors :-

(1) $\mathbf{x} = [\, x_1\ x_2\ \ldots .\ x_i\ x_j \ldots\ x_N\,]'$; $\mathbf{y} = [\, y_1\ y_2\ \ldots .\ y_m \ldots\ y_M\,]'$

The solution involves three major and quite discrete aspects :-

A - TOPOLOGY of the net, which in turn involves :

DEFINITION of the net - the labelling of the nodes and elements, and the statement of which elements are connected between what nodes.

ORIENTATION of the elements - the nomination of a positive flow sense through each element; as a result of this, one of the two nodes will be the inlet node to the element, the other becomes the exit.

CONSTRAINTS - the definition of the operable network laws; eg Kirchoff's Laws :

$\Sigma\, y_m = 0$ at any node, eg flow continuity - or its dual

$\Sigma\,(x_i - x_j) = 0$ around any closed loop.

Some observations about these laws include the following :

- one set only is necessary; we shall use only the *continuity* equations here
- the laws are intrinsically *linear*, eg no squares of the variables appear
- the formulation of the equations depends only on the *topology*, eg the equations for electric and fluid systems are identical.

B - CONSTITUTIVE LAWS which relate the across- and through-variables for the elements. These are known, ***application-dependent*** equations of the form :-

$x_i - x_j$ = function (y_m)

Eg, an electrical resistance : $v_i - v_j = R_m\, i_m$ Linear

or an hydraulic resistance : $h_i - h_j \approx R_m\, q_m^2$ Non-linear

The constitutive laws may not be algebraic equations, but differential or integral equations as occur in dynamic systems or inductor/capacitor nets. *Note the consequences of orientation when the constitutive laws are non-linear - what happens if $h_i < h_j$?*

C - SOLUTION OF EQUATIONS which result from combination of the constitutive and network laws. Although this might be thought the easiest step, especially if the network is linear, the desired accuracy may be difficult to achieve economically if the net is large.

Network analysis will be demonstrated only for linear systems of the type shown above, in which the elements are two-noded - that is the net can be regarded as a series of interconnected 1-dimensional lines. Such a net is known as a ***1-net***.

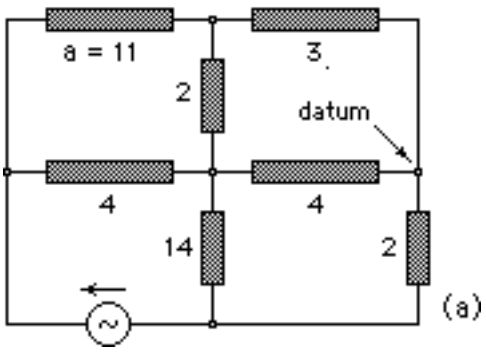
Analysis will first be carried out for a representative net using a familiar algebraic technique. This will give the background for a more general matrix approach which is the theoretical basis of a computer-oriented method for assembling the net equations - the *Direct Assembly* method.

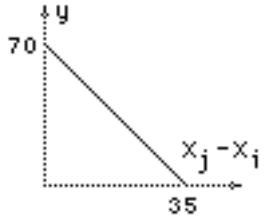
EXAMPLE Determine the across-variables, **x**, and the through-variables, **y**, for the typical network illustrated at (a) below. Constitutive laws are :-

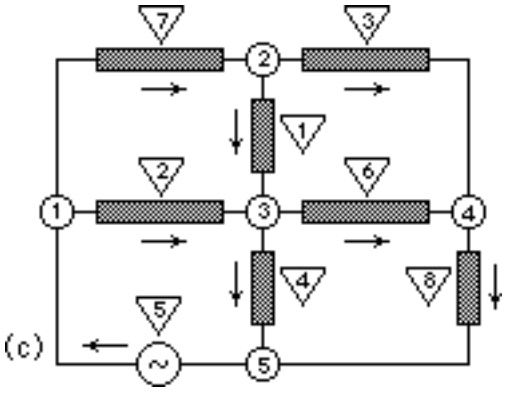
Passives $y_m = a_m\,(x_i - x_j)$ where the a-constants have the values shown in (a)

Active $y = 2\,(x_i - x_j) + 70$ a pump or battery with orientation as sketched ($x_i \le x_j$); also see (b).

The specified value of the across-variable is $x_0 = 5$ at the specified datum node.







Firstly the nodes, and then the elements, are numbered sequentially in any order and a positive sense nominated for the through-variable in each passive, ie the orientation of the net elements is completed, sketch (c).

The constitutive laws for all elements are written in an ordered manner, equations (A) below. Although the specified datum value $x_4 = 5$ is known, the value's substitution is delayed.

(A)

element	node in-out		node 1	2	3	4	5	
1	2 - 3	y_1 =	2(	$+x_2$	$-x_3$			)
2	1 - 3	y_2 =	4(	$+x_1$	$-x_3$			)
3	2 - 4	y_3 =	3(	$+x_2$		$-x_4$		)
4	3 - 5	y_4 =	14(		$+x_3$		$-x_5$	)
5	5 - 1	y_5 =	2(	$-x_1$			$+x_5$	) +70
6	3 - 4	y_6 =	4(		$+x_3$	$-x_4$		)
7	1 - 2	y_7 =	11(	$+x_1$	$-x_2$			)
8	4 - 5	y_8 =	2(			$+x_4$	$-x_5$	)

(B)

nodes	element 1 2-3	2 1-3	3 2-4	4 3-5	5 5-1	6 3-4	7 1-2	8 4-5	
1		$+y_2$			$-y_5$		$+y_7$		= 0
2	$+y_1$		$+y_3$				$-y_7$		= 0
3	$-y_1$	$-y_2$		$+y_4$		$+y_6$			= 0
4			$-y_3$			$-y_6$		$+y_8$	= 0
5				$-y_4$	$+y_5$			$-y_8$	= 0

(C)

element 1	2	3	4	5	6	7	8	
	$+4(x_1 - x_3)$			$+2(x_1 - x_5) - 70$		$+11(x_1 - x_2)$		= 0
$+2(x_2 - x_3)$		$+3(x_2 - x_4)$				$-11(x_1 - x_2)$		= 0
$-2(x_2 - x_3) - 4(x_1 - x_3) + 14(x_3 - x_5)$					$+4(x_3 - x_4)$			= 0
	$-3(x_2 - x_4)$				$-4(x_3 - x_4)$		$+2(x_4 - x_5)$	= 0
		$-14(x_3 - x_5) - 2(x_1 - x_5) + 70$				$-2(x_4 - x_5)$		= 0

(D)

	node	across-variable 1	2	3	4	5		
continuity	1	$17x_1$	$-11x_2$	$-4x_3$		$-2x_5$	=	70
	2	$-11x_1$	$+16x_2$	$-2x_3$	$-3x_4$		=	0
	3	$-4x_1$	$-2x_2$	$+24x_3$	$-4x_4$	$-14x_5$	=	0
	4		$-3x_2$	$-4x_3$	$+9x_4$	$-2x_5$	=	0
	5	$-2x_1$		$-14x_3$	$-2x_4$	$+18x_5$	=	-70

(E)

	node	across-variable 1	2	3	5		
continuity	1	$17x_1$	$-11x_2$	$-4x_3$	$-2x_5$	=	70
	2	$-11x_1$	$+16x_2$	$-2x_3$		=	15
	3	$-4x_1$	$-2x_2$	$+24x_3$	$-14x_5$	=	20
	4		$-3x_2$	$-4x_3$	$-2x_5$	=	-45
	5	$-2x_1$		$-14x_3$	$+18x_5$	=	-60

(F)

	node	across-variable 1	2	3	5		
continuity	1	$17x_1$	$-11x_2$	$-4x_3$	$-2x_5$	=	70
	2	$-11x_1$	$+16x_2$	$-2x_3$		=	15
	3	$-4x_1$	$-2x_2$	$+24x_3$	$-14x_5$	=	20
	5	$-2x_1$		$-14x_3$	$+18x_5$	=	-60

The continuity equations, (B), are written for all the nodes, ***preserving the same order of nodes and of elements as in the previous step***.

Any one of these equations (B) is redundant, but its deletion is delayed.

The constitutive equations (A) are inserted into the continuity equations (B), producing (C), whose across-variable terms are collected to yield the symmetric set of 5 continuity equations (D) in the 5 nodal variables.

The known datum, $x_4 = 5$, is substituted in (D) and the column taken to the RHS, to give the set (E), from which the superfluous continuity equation corresponding to the datum index (the fourth here) is eliminated, leaving the symmetric set of (N - 1) independent equations (F).

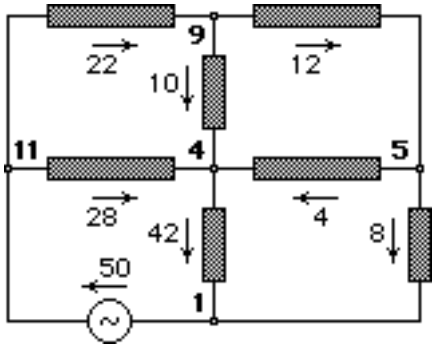
This is solved for the across-variables, yielding :-

$x_1 = 11$ $x_2 = 9$ $x_3 = 4$ ($x_4 = 5$) $x_5 = 1$

Inserting these values into the constitutive laws (A) for the through-variables gives :-

$y_1 = 10$ $y_2 = 28$ $y_3 = 12$ $y_4 = 42$
$y_5 = 50$ $y_6 = -4$ $y_7 = 22$ $y_8 = 8$

The initially assumed sense of the through-variable in element #6 was wrong; this is corrected in the sketch of the final solution below :-



This example is now repeated using matrix notation, assuming that the first step involving annotation and orientation has already been completed.

The constitutive laws of the M=8 elements, involving the N=5 nodes, are as follows. Zero matrix elements are represented by periods for legibility.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} 2 & . & . & . & . & . & . & . \\ . & 4 & . & . & . & . & . & . \\ . & . & 3 & . & . & . & . & . \\ . & . & . & 14 & . & . & . & . \\ . & . & . & . & 2 & . & . & . \\ . & . & . & . & . & 4 & . & . \\ . & . & . & . & . & . & 11 & . \\ . & . & . & . & . & . & . & 2 \end{bmatrix} \begin{bmatrix} . & +1 & -1 & . & . \\ +1 & . & -1 & . & . \\ . & +1 & . & -1 & . \\ . & . & +1 & . & -1 \\ -1 & . & . & . & +1 \\ . & . & +1 & -1 & . \\ +1 & -1 & . & . & . \\ . & . & . & +1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} + \begin{bmatrix} . \\ . \\ . \\ . \\ 70 \\ . \\ . \\ . \end{bmatrix}$$

or, in brief :

(**A**') **y** = **a t x** + $\mathbf{y}_o$

in which the unknown vectors, **x** and **y**, are as defined in (1), and the known matrices **a** and **t** and the vector $\mathbf{y}_o$ are defined to be :-

- **a** is an M∗M square diagonal matrix of admittance constants; all off-diagonal matrix elements are zero. The ordering of the constants in **a** is the same as that of the net elements in **y**.
- **t** is an M∗N topology incidence matrix, which identifies those two elements of the across-variable **x** which are relevant to each net element; so each row of **t** consists of a single +1 and a single -1, all remaining columns being zero.

$$\mathbf{a} = \begin{bmatrix} 2 & . & . & . & . & . & . & . \\ . & 4 & . & . & . & . & . & . \\ . & . & 3 & . & . & . & . & . \\ . & . & . & 14 & . & . & . & . \\ . & . & . & . & 2 & . & . & . \\ . & . & . & . & . & 4 & . & . \\ . & . & . & . & . & . & 11 & . \\ . & . & . & . & . & . & . & 2 \end{bmatrix} \quad \mathbf{t} = \begin{bmatrix} . & +1 & -1 & . & . \\ +1 & . & -1 & . & . \\ . & +1 & . & -1 & . \\ . & . & +1 & . & -1 \\ -1 & . & . & . & +1 \\ . & . & +1 & -1 & . \\ +1 & -1 & . & . & . \\ . & . & . & +1 & -1 \end{bmatrix} \quad \mathbf{y}_o = \begin{bmatrix} . \\ . \\ . \\ . \\ 70 \\ . \\ . \\ . \end{bmatrix}$$

- $\mathbf{y}_o$ is an M-vector of specified through-variables, with non-zero elements corresponding to the network active elements, or sources.

The N = 5 nodal continuity equations are

(**B**') **t**' **y** = **0** or, in full :-

$$\begin{bmatrix} . & +1 & . & . & -1 & . & +1 & . \\ +1 & . & +1 & . & . & . & -1 & . \\ -1 & -1 & . & +1 & . & +1 & . & . \\ . & . & -1 & . & . & -1 & . & +1 \\ . & . & . & -1 & +1 & . & . & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} . \\ . \\ . \\ . \\ . \end{bmatrix}$$

in which the **y** ***pre-multiplier is just the transpose of the topology matrix*** **t**, due to the equations being written down in an orderly manner. Combining the two sets of equations as was done algebraically above, by inserting (A') into (B') leads to : **t**' (**a t x** + $\mathbf{y}_o$) = **0** or, in short :

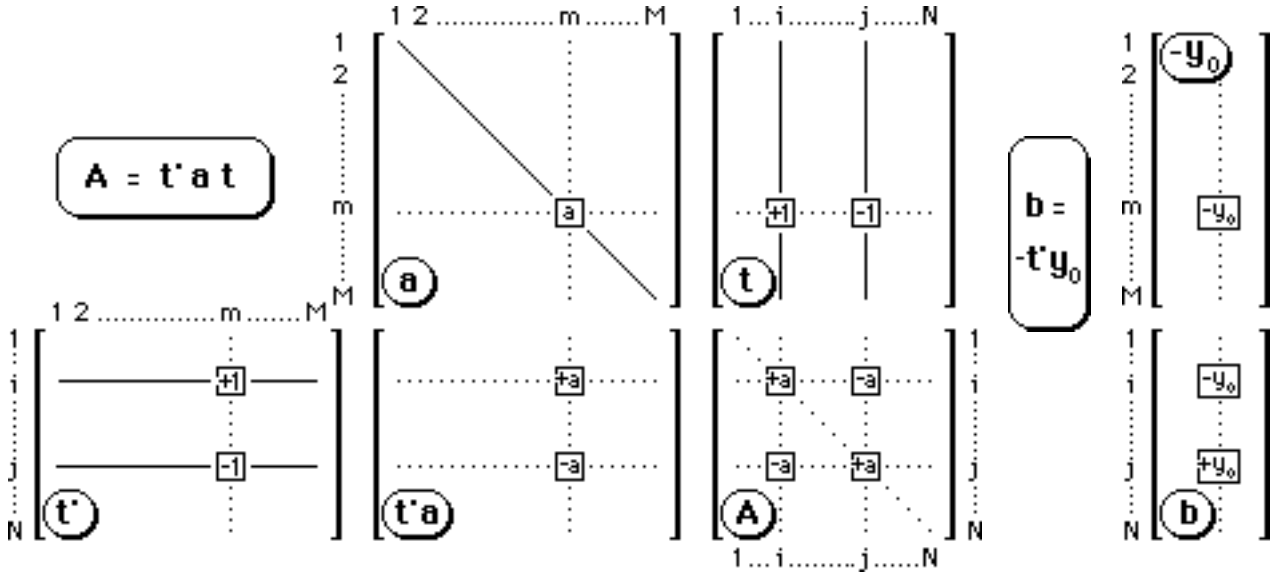
(**2**) **A x** = **b** where **A** = **t**' **a t** and **b** = -**t**' $\mathbf{y}_o$

This matrix equation is identical to the algebraic equations (D) above. Once it is reduced by the datum across-variable and the corresponding redundant continuity equation, it can be solved by standard procedures, since both **A** and **b** are known.

Strict matrix algebra is seldom used in practical applications since it demands too much unnecessary storage, and computations take too long with the sparse matrices which usually occur. The matrix approach is useful however in demonstrating how the transpose triple product **A**, and **b**, may be assembled directly in the computer. Thus, consider the contribution to **A** and to **b**

of the m'th element, for which the following are known (being held typically in a data file) :

- the inlet node index 'i', and outlet node index 'j',
- the admittance constant 'a', and
- the specified through-variable, y_o (zero for a passive element)



It can be seen that the overall result of these detailed matrix multiplications is equivalent to merely adding / subtracting the element data to / from the system equations.

The direct assembly process is therefore :

- null all elements of the N∗N **A**-matrix and of the N-vector, **b**;
- for each network element in turn, read the above data from file and :
  - add the a-constant to the (i, i)th and (j, j)th diagonal elements of **A**, and
  - subtract it from the (i, j)th and (j, i)th off-diagonal elements of **A**;
  - subtract y_o from the ith element of **b**, and add it to the jth element.

Carrying out this process for the example above :-

Data file

m	i-j	a	y_o
1	2-3	2	.
2	1-3	4	.
3	2-4	3	.
4	3-5	14	.
5	5-1	2	70
6	3-4	4	.
7	1-2	11	.
8	4-5	2	.

Direct assembly of **A** and **b**

A	1	2	j	3	4	5	**b**
1	+11 +2 +4	-11	-4	·	-2		70
2	-11	+11 +3 +2	-2	-3	·		·
i							
3	-4	-2	+4 +14 +4 +2	-4	-14		·
4	·	-3	-4	+2 +4 +3	-2		·
5	-2	·	-14	-2	+2 +2 +14		-70

Equations **A x = b** - as assembled

$$\begin{bmatrix} 17 & -11 & -4 & . & -2 \\ -11 & 16 & -2 & -3 & . \\ -4 & -2 & 24 & -4 & -14 \\ . & -3 & -4 & 9 & -2 \\ -2 & . & -14 & -2 & 18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 70 \\ . \\ . \\ . \\ -70 \end{bmatrix}$$

- and finally, as reduced

$$\begin{bmatrix} 17 & -11 & -4 & . & -2 \\ -11 & 16 & -2 & . & . \\ -4 & -2 & 24 & . & -14 \\ . & . & . & 1 & . \\ -2 & . & -14 & . & 18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} 70 \\ 15 \\ 20 \\ 5 \\ -60 \end{bmatrix}$$

Reduction to the final equations here consists of substituting for the datum value and transposing the corresponding terms to the RHS, then, rather than eliminating the relevant continuity equation, it is more simply overwritten by the expression for the datum across-variable. This preserves symmetry and requires storage of half the **A** matrix only. It is noticeable that the element input data does not have to be stored but is used immediately it is read from file.

The output of the Pascal program 'Linet' for analysing linear networks appears below, together with the data file for the foregoing example. The program uses the direct assembly technique, incorporates the simultaneous equation solver 'Simeq', automatically accumulates the total

DATA FILE : LINX1

```
LINEAR NETWORKS EXAMPLE 1
4 5          (datum node,value)
2 3   2  0   (first element)
1 3   4  0   (second element)
2 4   3  0   ......
3 5  14  0   (in#,out#,a,y0)
5 1   2 70   ......
3 4   4  0   ......
1 2  11  0   ......
4 5   2  0   (last element)
```

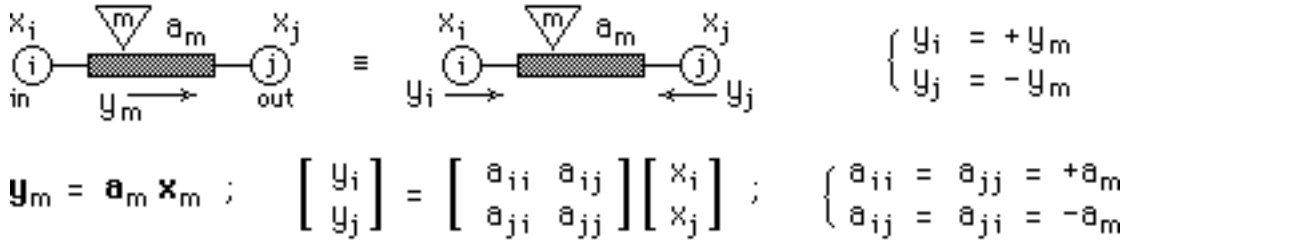
number of elements and of nodes, avoids internal storage of element data but echoes it together with the problem title as a visual check, orients correctly the node indices on output (eg element #6) and includes as a check on numerical accuracy the theoretically-zero nodal continuity excesses.

```
**************        Linear
PROGRAM LINET       1-Network
**************        Analysis

Enter data file name : LINX1

LINEAR NETWORKS EXAMPLE 1

Elm't  in-      y       a      y0
index  out
  1    2-3    10.0     2.0     0
  2    1-3    28.0     4.0     0
  3    2-4    12.0     3.0     0
  4    3-5    42.0    14.0     0
  5    5-1    50.0     2.0    70
  6    4-3     4.0     4.0     0
  7    1-2    22.0    11.0     0
  8    4-5     8.0     2.0     0

Node    x    continuity
index          excess
  1   11.0    -1.1e-5
  2    9.0     0.0e 0
  3    4.0     1.1e-5
  4    5.0    -5.7e-6
  5    1.0     9.5e-6

End of program
```

Network analysis usually involves a transpose triple product akin to (2); non-linear nets can often be linearised into this form. Direct assembly as above is a useful technique for such situations but it is not always the most efficient, since in large systems the bandwidth requirements may be crucial, as will be shown later.

Extension to 2- and 3-Networks

In order to generalise the foregoing to more complex networks, we first redefine **y** as a nodal variable akin to **x**. That is, **x** and **y** are dual or complementary sets of field variables measured at the nodes of the network, and the concept of element orientation becomes irrelevant. If, in the flow net above, y is reckoned as a nodal flow ***towards*** the element in question, the constitutive law for the m'th element may be written in matrix form as below :



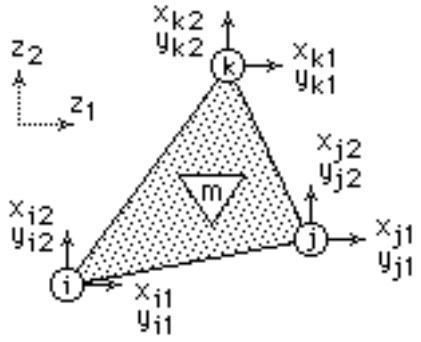
$$\mathbf{y}_m = \mathbf{a}_m \mathbf{x}_m \;; \quad \begin{bmatrix} y_i \\ y_j \end{bmatrix} = \begin{bmatrix} a_{ii} & a_{ij} \\ a_{ji} & a_{jj} \end{bmatrix} \begin{bmatrix} x_i \\ x_j \end{bmatrix} \;; \quad \begin{cases} a_{ii} = a_{jj} = +a_m \\ a_{ij} = a_{ji} = -a_m \end{cases}$$

Direct assembly is thus seen to be the <u>addition</u> of the relevant sub-matrices of the element's a-matrix to the corresponding locations in the system's A-matrix.

In the nets above, there is one degree of freedom at each node (ie both 'x' and 'y' are scalar) and so the sub-matrices of the element **a**-matrix are also scalar.

These ideas may easily be expanded to more complex networks in which the concept of **y** as a flow is unnecessarily restrictive and will now be dropped - **x** and **y** are just a pair of complementary nodal variables. Thus **x** may be interpreted as nodal displacement and **y** as nodal force, in which case **a** and **A** correspond to element and system stiffnesses respectively, and (2) are equations of equilibrium - <u>we shall concentrate on this interpretation in what follows</u>; however, it is possible to reverse the roles of **x** and **y**, in which case (2) are compatibility equations and **A** and **a** are flexibilities. This conjugality between equilibrium and compatibility, like that already noted between the two Kirchoff's Laws, is a consequence of the fundamental duality of networks; any given net can be transformed into its dual net by interchanging nodes with elements.

Consider the triangular element in two dimensions (z_1, z_2) illustrated below, the nodes being labelled 'i', 'j', 'k'. There are two components of both 'x' and 'y' at each node, so the constitutive law for the m'th element may be written variously as shown - in the full 6*6 scalar form (i), or



$$\textbf{(i)}\quad \begin{bmatrix} y_{i1} \\ y_{i2} \\ \cdots \\ y_{j1} \\ y_{j2} \\ \cdots \\ y_{k1} \\ y_{k2} \end{bmatrix} = \begin{bmatrix} a_{ii11} & a_{ii12} & a_{ij11} & a_{ij12} & a_{ik11} & a_{ik12} \\ & a_{ii22} & a_{ij21} & a_{ij22} & a_{ik21} & a_{ik22} \\ & & a_{jj11} & a_{jj12} & a_{jk11} & a_{jk12} \\ & & & a_{jj22} & a_{jk21} & a_{jk22} \\ & & & & a_{kk11} & a_{kk12} \\ \text{Symmetric} & & & & & a_{kk22} \end{bmatrix} \begin{bmatrix} x_{i1} \\ x_{i2} \\ x_{j1} \\ x_{j2} \\ x_{k1} \\ x_{k2} \end{bmatrix}$$

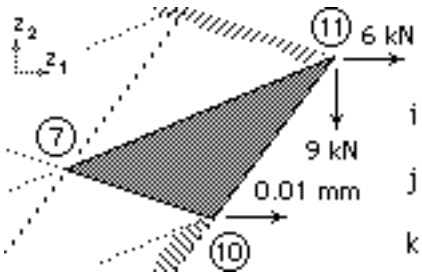
$$\textbf{(ii)}\quad \begin{bmatrix} \mathbf{y}_i \\ \mathbf{y}_j \\ \mathbf{y}_k \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{ii} & \mathbf{a}_{ij} & \mathbf{a}_{ik} \\ & \mathbf{a}_{jj} & \mathbf{a}_{jk} \\ \text{Sym} & & \mathbf{a}_{kk} \end{bmatrix} \begin{bmatrix} \mathbf{x}_i \\ \mathbf{x}_j \\ \mathbf{x}_k \end{bmatrix} ; \quad \mathbf{a}_{pq} = \begin{bmatrix} a_{pq11} & a_{pq12} \\ \text{Sym} & a_{pq22} \end{bmatrix}$$

$$\textbf{(iii)}\quad \mathbf{y}_m = \mathbf{a}_m \mathbf{x}_m \quad ; \quad \mathbf{a}_m = \begin{bmatrix} \mathbf{a}_{ii} & \mathbf{a}_{ij} & \mathbf{a}_{ik} \\ & \mathbf{a}_{jj} & \mathbf{a}_{jk} \\ \text{Sym} & & \mathbf{a}_{kk} \end{bmatrix}$$

in shorthand vector notation (iii) where the **a**-matrix consists of 3∗3 sub-matrices corresponding to the three nodes, each of the nine sub-matrices being itself 2∗2 (ii) corresponding to the two degrees of freedom. Evaluation of the various a-scalars is an application-dependent detail, however no matter what the application (provided it involves the transpose triple product) the **A**-assembly process is identical to that outlined above.

b on the RHS of (2) is just a vector of specified y-values - that is, since we are interpreting **y** as force, of defined concentrated nodal loads. Assembly of **b** is now simpler than it was in the case of flow nets; in that case, sources were added to one element of **b** and subtracted from another - a specified-y affected two nodes. In the present case however, (2) are equations of nodal equilibrium, the specified-y's are known external forces at single nodes of the system, and orientation as considered previously is irrelevant. So the external loads are merely inserted individually into **b**. Thus¡ :

EXAMPLE The triangular lamina is located at the corner of the assemblage illustrated, and its stiffness matrix is quoted in MN/mm units. Investigate its contribution to the equilibrium equations of the assemblage.



$$\begin{array}{c|cc|cc|cc} & i=10 & & j=11 & & k=7 & \\ & 1 & 2 & 1 & 2 & 1 & 2 \\ \hline i\,\{1 & 2 & 3 & 3 & 1 & -2 & 0 \\ \quad\{2 & & 4 & 7 & 6 & -2 & 6 \\ \hline j\,\{1 & & & 5 & -3 & 2 & 1 \\ \quad\{2 & & & & 2 & 5 & -4 \\ \hline k\,\{1 & & & & & 3 & 4 \\ \quad\{2 & \text{Sym} & & & & & 1 \end{array}$$

The element's stiffness sub-matrices are assembled directly into the system **A**-matrix in the sketch below, remembering that all terms are additive to existing coefficients and do not overwrite them.

The specified force components at node 11 are simply substituted into the **b**-vector.

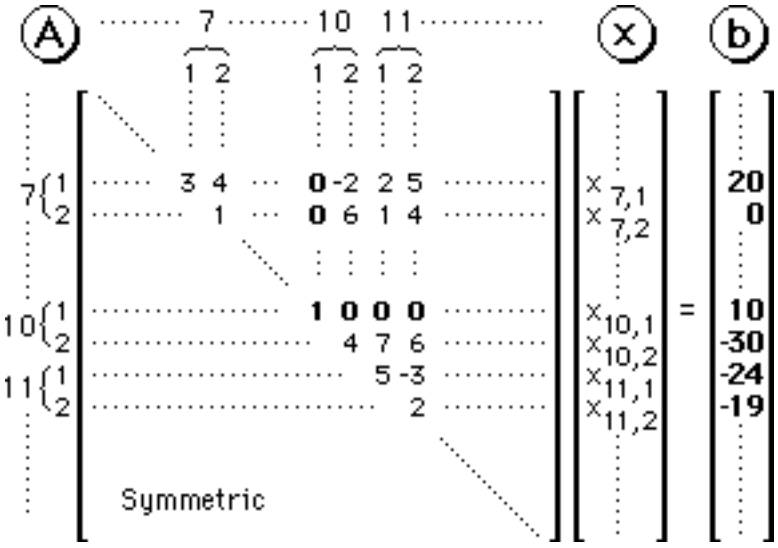
When all elements have been similarly assembled into the system equations, any specified displacements, x, are evaluated and extracted to the RHS, and the corresponding equilibrium equation overwritten by the expressions for the specified values - exactly as was done for the 1-networks above.

Support reactions would thus be defined as points having no displacement. Substitution of

$$\textbf{(a)}\quad \begin{array}{c|cc|cc|cc} & 10 & & 11 & & 7 & \\ & 1 & 2 & 1 & 2 & 1 & 2 \\ \hline 10\,\{1 & +2 & +3 & +3 & +1 & -2 & +0 \\ \quad\{2 & +3 & +4 & +7 & +6 & -2 & +6 \\ \hline 11\,\{1 & +3 & +7 & +5 & -3 & +2 & +1 \\ \quad\{2 & +1 & +6 & -3 & +2 & +5 & -4 \\ \hline 7\,\{1 & -2 & -2 & +2 & +5 & +3 & +4 \\ \quad\{2 & +0 & +6 & +1 & -4 & +4 & +1 \end{array}$$

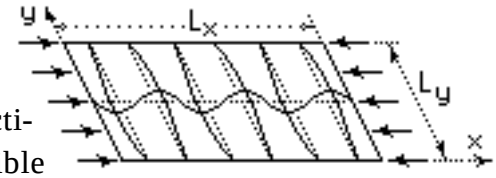
$$\textbf{(A)}\quad \begin{array}{c|cc|cc|cc} & 7 & & 10 & & 11 & \\ & 1 & 2 & 1 & 2 & 1 & 2 \\ \hline 7\,\{1 & +3 & +4 & -2 & -2 & +2 & +5 \\ \quad\{2 & +4 & +1 & +0 & +6 & +1 & -4 \\ \hline 10\,\{1 & -2 & +0 & +2 & +3 & +3 & +1 \\ \quad\{2 & -2 & +6 & +3 & +4 & +7 & +6 \\ \hline 11\,\{1 & +2 & +1 & +3 & +7 & +5 & -3 \\ \quad\{2 & +5 & -4 & +1 & +6 & -3 & +2 \end{array} \quad \textbf{(x)} \quad \begin{bmatrix} x_{7,1} \\ x_{7,2} \\ x_{10,1} \\ x_{10,2} \\ x_{11,1} \\ x_{11,2} \end{bmatrix} = \textbf{(b)} \quad \begin{bmatrix} \\ \\ \\ \\ 6 \\ -9 \end{bmatrix}$$

$x_{10,1}$ = 0.01 mm is shown in the matrix here, neglecting for demonstration simplicity the contributions to **A** of elements other than the present one of interest.



Once the equilibrium equations have been solved, the nodal displacements are substituted back into the elements' constitutive laws - again, just like the simple 1-nets above - to determine nodal forces. We shall see later how stresses also can be derived.

Extension to three-dimensions evidently requires 3∗3 sub-matrices of **a** and **A** - their assembly is readily visualised - but how do we determine an element stiffness matrix to assemble in this manner ? Before we demonstrate this, we shall introduce the Rayleigh-Ritz method - a powerful general technique for structural analysis, not restricted to Finite Elements.



The Rayleigh-Ritz Method

The edge-supported buckled rectangular plate is a typical practical component of simple unloaded shape where it is impossible to deduce the true, closed-form stresses. To obtain some idea of the stress levels therefore, an approximation to the deformations is first postulated. This mathematical function, the 'displacement model', reflects the form or general shape of the deformations, and it is usually a series incorporating a number of undetermined coefficients. Although the inter-relation between these is initially chosen so that the model satisfies compatibility (eg the geometric boundary conditions), this necessity is insufficient for the complete evaluation of the coefficients - they must be quantified later to complete the model.

The out-of-plane model displacement of the above plate, u (scalar), might here be taken as a truncation of the Fourier series :-

$$u\{x,y\} = \sum_{m=1} a_m \sin m\pi x/L_x . \sin \pi y/L_y$$ where the constant a_m's are as yet unknown.

This displacement model automatically satisfies the boundary condition around the edges, no matter what values are assumed by the a-coefficients, or at what harmonic the model is truncated. Truncated power series - again with undetermined constant coefficients - are superior to Fourier series for certain other components. Obviously, the greater the number of terms, the better the model's accuracy (provided the exact solution is not contained in the series.)

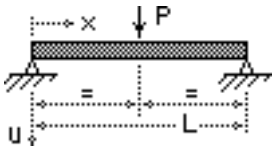
Having laid down the form of the model, the undetermined coefficients are evaluated afterwards by considering equilibrium. We have employed this technique in the past for stresses in shafts, in beams, both straight and curved, and in fillet welds - however in these cases the assumed deformations were the correct ones and so the equilibrium equations also were exact. If deformations are approximate however, the equilibrium equations which are obtained from them (by differentiation) are also approximate - in fact since differentiation is an error-magnifying process, derived reactions and stresses are less accurate than the displacements. In view of the approximate nature of the equilibrium equations, they are usually gotten from the Principle of Minimum Potential Energy rather than from elementary Statics.

The method may be demonstrated by the simple beam shown overleaf, whose exact solution may be used as a yardstick. A candidate model for the deflection might be either :

$u = a_1 \sin \pi x/L + a_2 \sin 2\pi x/L + a_3 \sin 3\pi x/L + \ldots\ldots$ or

$u = b_0 + b_1 x + b_2 x^2 + b_3 x^3 + b_4 x^4 + \ldots\ldots$

where the a's and b's are constants, as yet unknown.



A single Fourier term will be chosen here for example; it is evident that a_1 is the maximum displacement and that the model meets the geometric boundary conditions automatically :-

$u = u_{max} \sin \pi x/L$ (satisfying :- $u = 0$ @ $x = 0, L$ and $u = u_{max}$ @ $x = L/2$)

It may be shown that the strain energy of a linear elastic beam may be expressed variously as :-

$\Pi_{beam} = {}^1/_{2EI} \int_L M^2 \, dx$ or $= {}^{EI}/_2 \int_L (u'')^2 \, dx$ the second formulation is relevant here

so the potential of the system becomes

$\Pi = \Pi_{beam} + \Pi_{load} = \pi^4 EI \, u^2_{max} / 4L^3 - P.u_{max}$

ie, a function of the undetermined coefficients, of which there is only one here, u_{max}.

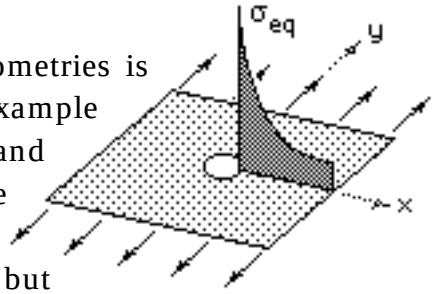
The Principle of Minimum Potential Energy stipulates that, for equilibrium :-

$\partial\Pi / \partial u_{max} = 0$ ie, here $(\pi^4 EI / 2L^3) \, u_{max} - P = 0$

This last equation is clearly an equilibrium equation, of the form $K.u_{max} = P$, where K is the (approximate) stiffness of the beam as seen by the load, P. The equation predicts that $u_{max} = 2PL^3 / \pi^4 EI = PL^3 / 48.7 \, EI$, which compares favourably with the exact expression (48 in the denominator). Evaluation of the unknown displacement coefficient(s) by the above technique enables completion of the displacement model, u. The stress resultant follows, from beam theory

$M = - EI \, u'' = EI \, u_{max} \, (\pi/L)^2 \sin \pi x/L = (2/\pi^2) \, PL \sin \pi x/L = 0.203 \, PL \sin \pi x/L$

As pointed out above, this approximation is not as satisfactory as that of the displacements - the exact variation is linear with a central moment of 0.25 PL.



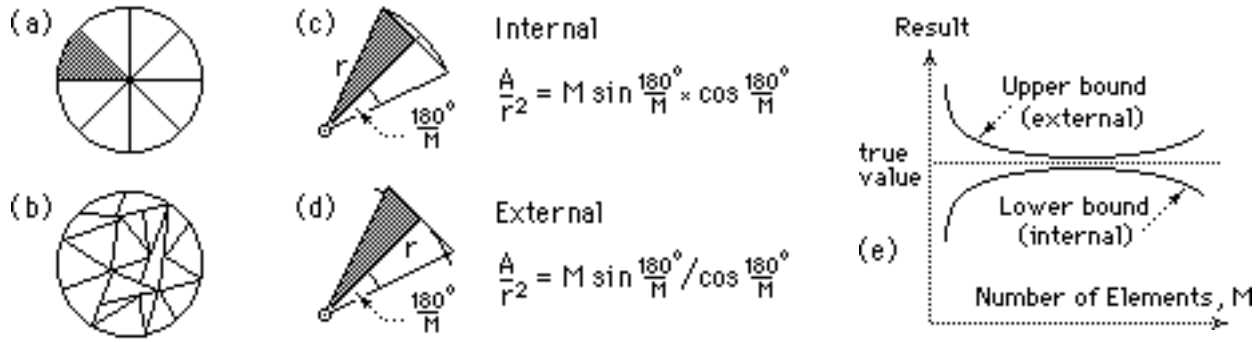
Application of the Rayleigh-Ritz method to complex unloaded geometries is often impractical - the displacement model for a holed plate for example has to account for the discontinuity at the hole's circumference and also has to include a large number of terms to adequately describe the gross non-linearities. This is not impossible, since geometric transformations are available to map the plate into a rectangle - but the resulting model is far too cumbersome for routine work. This is where the concept of Finite Elements comes into its own. The body is divided into a number of contiguous elements which are laid out to describe its approximate geometry, as in the frontispiece; body integrals become

$U_{body} = \int_{\text{over the whole body}} = \sum_{\text{over all elements in the body}} \left(\int_{\text{over one element}} \right)$

The displacement model now only needs to be applied to the individual elements - whose shapes are simple - and the model itself can be relatively crude since parameters which vary steeply over the body do not vary so markedly over each individual element. Accuracy may apparently be enhanced by elements which are either more numerous or more complex, but generally all elements in an assemblage should be based on the same displacement model.

The following example shows some further features of the FEM. It is required to estimate the value of π by approximating the area of a circle of radius 'r', by triangles. The area is subdivided *rationally* into 'M' sectors as shown at (a) below for M=8. Refining the net by doubling the number of elements is straightforward, unlike the haphazard net of (b) which contains also four- and five-sided elements - nets should be set up with ease of possible subdivision in mind.

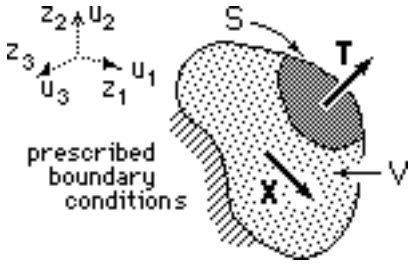
The sectors of (a) are approximated by triangles, either internal as shown at (c), or external as at (d), and the total area of the triangles evaluated in each case. The graph of these results (e) shows



some trends which are generally true of FEM analyses :-

- As the network is refined, ie as the number of elements increases, the results become more accurate - converge to the true value - until swamped by computational inaccuracies when the equation set becomes large.
- When the net is ***reducible***, and the other techniques which are used are correct, then we may be fairly certain that results are ***bounds*** to the true value. A reducible net is one which is contained completely by the previous coarser net as refinement proceeds. Bounding is important theoretically, although not routinely used for practical problems since confidence in the FEM has been established. It can be achieved by using the dual (again !) energy theorems, for example; or the dual stiffness and flexibility methods of analysis.

As noted, accuracy may be improved by using elements which are *more complex* individually, rather than *more numerous*. Thus if we had attempted the estimation above based upon circumference rather than on area, then, for a given result with a certain number of straight line elements, we might have achieved the same accuracy using fewer second order curved line elements (if we had them). If a circular element were available, then of course only one would be required for the 'correct' solution. Analysis economics in practice thus requires a trade-off between many simple elements on the one hand, where computation time is devoted mainly to assembling and solving a large number of simple equations, or relatively few elements having complex constitutive laws on the other, in which case the constitutive laws of the individual elements form the bulk of computation costs. The cost of setting up a model for a given prototype will increase with the number of elements.

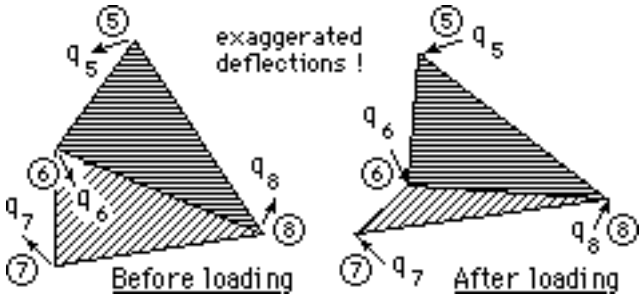


Finite Element Theory - Equilibrium of the Discretised Body

In presenting an outline of the finite element method in this section, no attempt is made to be rigorous or to explain every detail; the purpose is solely to give a general broad-brush appreciation of the method.

An elastic body of known geometry is defined in three-dimensional **z**-space and subjected to the body force **X** throughout its volume, V. Prescribed tractions **T** (ie loads) are applied over part of the surface S of the body's boundary, and a further portion of the boundary is subjected to prescribed displacements (eg supported).

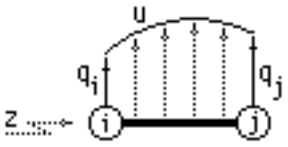
The body is discretised into a system of 'M' contiguous elements, interconnected at their nodes. The displacement of the body is described by the ***vector of discrete nodal displacments,*** $\mathbf{q} = [\ldots \mathbf{q}_i\ \mathbf{q}_j \ldots]'$.



The ***continuous displacement field,*** $\mathbf{u}\{\mathbf{z}\} = [\ u_1\ u_2\ u_3\ ]'$, over any particular element must be compatible with the corresponding nodal displacements of

that element - ie with a sub-vector of **q**. The displacement model for the element in question can thus be expressed as :-

(3) $\mathbf{u} = \mathbf{N}\,\mathbf{q}$ where the 'shape function' $\mathbf{N}\{\mathbf{z}\}$, is a continuous function of location.



We shall examine the implications of this in more detail in the next section, but in order to fix our ideas for the time being, consider a one-dimensional element, located between nodes 'i' and 'j' of the assemblage. Displacement is scalar, and we may postulate the second order model : $u = a_0 + a_1 z + a_2 z^2$ involving three undetermined coefficients. But compatibility requires that $u = q_i$ at z_i and $u = q_j$ at z_j, so that the model must be along the lines :

$u = q_i\,(z_j - z)/(z_j - z_i) + q_j\,(z - z_i)/(z_j - z_i) + a\,(z - z_i)/(z_j - z)$ in which 'a' is the sole coefficient open to arbitrary choice. It is evident that the number of coefficients must not be less than the total degrees of freedom of 'external' nodal displacements, two in this case, otherwise compatibility cannot possibly be achieved. External nodes connect an element to the rest of the assemblage; it is quite in order to define 'internal' nodes which are intrinsic to the element and are not connected to anything else.

If 'a' is chosen to be zero for the element in question, then the linear model is :

$$u = q_i(z_j - z)/(z_j - z_i) + q_j(z - z_i)/(z_j - z_i) = (1/(z_j - z_i))\,\mathbf{[}\,(z_j - z)\;\;(z - z_1)\,\mathbf{]}\,\mathbf{[}\,q_i\;q_j\,\mathbf{]}'$$

which is of the form (3) with the continuous shape function : $\mathbf{N} = (1/(z_j - z_i))\,\mathbf{[}\,(z_j - z)\;\;(z - z_1)\,\mathbf{]}$

Stress and strain will be characterised simply here by principals, thus $\sigma = \mathbf{[}\,\sigma_1\,\sigma_2\,\sigma_3\,\mathbf{]}'$ and $\varepsilon = \mathbf{[}\,\varepsilon_1\,\varepsilon_2\,\varepsilon_3\,\mathbf{]}'$; a more rigourous treatment incorporates shear in a similar manner. From elementary elasticity, the continuous geometric entities strain and displacement are inter-related via :

$\varepsilon_1 = \partial u_1 / \partial z_1$ and similarly for other components. Or, in brief :

$\varepsilon = \partial\,\mathbf{u}$ where ∂ is a matrix of partial differential operators.

From the above it follows that the strain also must depend upon the nodal displacement vector:

$\varepsilon = \partial\,\mathbf{N}\,\mathbf{q} = \mathbf{B}\,\mathbf{q}$ where the continuous strain-nodal displacement matrix, $\mathbf{B} = \partial\,\mathbf{N}$.

Furthermore, since the body is elastic, stresses are linearly related to strains since :

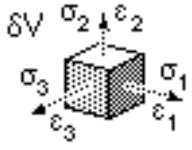
$\varepsilon_1 = (\,\sigma_1 - \nu\sigma_2 - \nu\sigma_3\,)/E$ etc, and so, solving for the principal stresses :

$\sigma_1 = \big(\,(1-\nu)\varepsilon_1 + \nu\varepsilon_2 + \nu\varepsilon_3\,\big)\,E\,/\,(1+\nu)(1-2\nu)$ etc; or, briefly :

$\sigma = \mathbf{C}\,\varepsilon = \mathbf{C}\,\mathbf{B}\,\mathbf{q}$ in which **C** is a matrix of material elastic constants.

Once an element's displacement model has been defined, **N** and **B** may be computed; then, when **q** is evaluated, these constitutive laws above may be recalled to ascertain stress and strain.

The work/energy terms relevant to the system consisting of an element and its associated loads are as follows :



- The gain in strain energy of the element, $\Delta\Pi_\varepsilon$, over the element's volume. Characterising stress and strain in the volume δV of the element by their principals, then $\Delta\Pi_\varepsilon = \int_V \tfrac{1}{2}\,(\,\varepsilon_1\sigma_1 + \varepsilon_2\sigma_2 + \varepsilon_3\sigma_3\,)\,dV = \int_V \tfrac{1}{2}\,\varepsilon'\sigma\,dV$.
- The work done by body forces such as weight, $\Delta\Pi_X$, over the element's volume. If the displacement and (constant) body force at the volume δV of the element are $\mathbf{u} = \mathbf{[}\,u_1\,u_2\,u_3\,\mathbf{]}'$ and $\mathbf{X} = \mathbf{[}\,X_1\,X_2\,X_3\,\mathbf{]}'$ respectively, then $\Delta\Pi_X = \int_V \mathbf{u}'\mathbf{X}\,dV$.
- The work done by tractions, $\Delta\Pi_T$, over the element's surface area. In a manner analogous to body force work, $\Delta\Pi_T = \int_S \mathbf{u}'\mathbf{T}\,dS$. If the element's surface is not part of the assembly's surface which is subject to traction, then this work term will vanish.
- The work done by other elements' contacts at the element's nodes, $\Delta\Pi_N$.

The gain in potential of the element and its loads is therefore :

$$\Delta\Pi = \Delta\Pi_\varepsilon - \Delta\Pi_X - \Delta\Pi_T - \Delta\Pi_N$$

$$= \int_V {}^1/_2\, \varepsilon' \sigma \, dV - \int_V \mathbf{u}' \mathbf{X} \, dV - \int_S \mathbf{u}' \mathbf{T} \, dS - \Delta\Pi_N$$

or, substituting from above for **u**, ε and σ :

$$= \int_V {}^1/_2\, \mathbf{q}' \mathbf{B}' \mathbf{C} \mathbf{B} \mathbf{q} \, dV - \int_V \mathbf{q}' \mathbf{N}' \mathbf{X} \, dV - \int_S \mathbf{q}' \mathbf{N}' \mathbf{T} \, dS - \Delta\Pi_N$$

Summing this over all elements in the assemblage, the potential gain of the body and its associated loading is :

$$\Pi = \Sigma_M \{ \int_V {}^1/_2\, \mathbf{q}' \mathbf{B}' \mathbf{C} \mathbf{B} \mathbf{q} \, dV - \int_V \mathbf{q}' \mathbf{N}' \mathbf{X} \, dV - \int_S \mathbf{q}' \mathbf{N}' \mathbf{T} \, dS \}$$

Note that the overall effect of $\Sigma_M \Delta\Pi_N$ must be zero, due to equal-and-opposite action/reaction contributions at the inter-element nodes. Since **q** is a vector of discrete nodal displacements, common to all elements and not subject to integration, it is extracted from the integrals to give :

$$\Pi = \mathbf{q}' \Sigma_M \{ (\int_V {}^1/_2\, \mathbf{B}' \mathbf{C} \mathbf{B} \, dV) \mathbf{q} - \int_V \mathbf{N}' \mathbf{X} \, dV - \int_S \mathbf{N}' \mathbf{T} \, dS \}$$

Applying the Principle of Minimum Potential Energy, equilibrium of the body requires that $\partial\Pi / \partial\mathbf{q} = 0$, or :

$$\{ \Sigma_M (\int_V \mathbf{B}' \mathbf{C} \mathbf{B} \, dV) \} \mathbf{q} = \Sigma_M (\int_V \mathbf{N}' \mathbf{X} \, dV + \int_S \mathbf{N}' \mathbf{T} \, dS) \quad \text{ie}$$

(4a) $$(\Sigma_M \mathbf{k}) \mathbf{q} = \Sigma_M \mathbf{p}$$

where $\mathbf{k} = \int_V \mathbf{B}' \mathbf{C} \mathbf{B} \, dV$ is the stiffness of an element, and

$\mathbf{p} = \int_V \mathbf{N}' \mathbf{X} \, dV + \int_S \mathbf{N}' \mathbf{T} \, dS$ is the external force on an element.

Simplifying this still further, the equilibrium equations for the discretised body are :

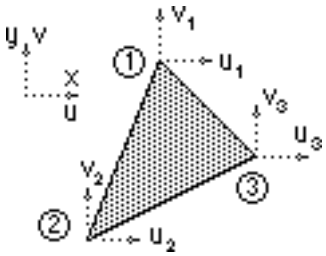
(4b) $$\mathbf{K}\mathbf{q} = \mathbf{P}$$

where $\mathbf{K} = \Sigma_M \mathbf{k}$ is the stiffness of the discretised body system, and

$\mathbf{P} = \Sigma_M \mathbf{p}$ is the external force vector acting on it.

The equilibrium equations (4b) are analagous to the network equations (2), and may be assembled identically since (4a) demonstrates that elemental stiffness involves the transpose triple product **B' C B**. The equations only have to be reduced by the prescribed displacements before being solved for **q** in the same way as (2) were solved previously for **x**. Having found **q**, the solution is completed by calculating stresses and strains from the constitutive laws above.

<u>Finite Element Theory - Element Stiffness</u>

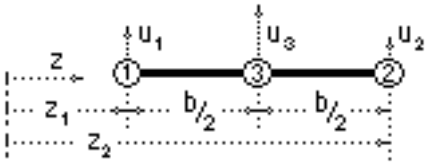


We have seen that knowledge of the strain-displacement matrix, **B**, is a fundamental necessity for evaluating element stiffnesses from (4a). We shall now illustrate typical approaches to finding **B**, and particularise on a simple element - the constant strain triangle used in the package, 'FEM1'.

The first step is the choice of a suitable shape function, **N** - that is, an equation which approximates the displacement, **u**, at any point in the element under consideration, in terms of :

- the co-ordinates of the point in question, and
- the displacement of the element's nodes, **q**, eg in the case of the 3-node, 2-dof element shown, either $\mathbf{q} = [\, u_1\, v_1\, u_2\, v_2\, u_3\, v_3 \,]'$ or $\mathbf{q} = [\, u_1\, u_2\, u_3\, v_1\, v_2\, v_3 \,]'$

There are two distinct methods of establishing this function - using either 'generalised coordinates' or 'natural coordinates'.



Generalised Co-ordinates

Consider a one-dimensional element of length 'b', with two external nodes, '1' & '2', and a central internal node '3'. There is one degree of freedom in (transverse) displacement, u. We take the displacement model in

power series form with constant coefficients, α_i :-

$u = \alpha_1 + \alpha_2 z + \alpha_3 z^2$ or, in brief

(a) $\mathbf{u} = \phi' \alpha$

where $\phi = [\ 1 \quad z \quad z^2\ ]'$ is a vector of powers of the coordinate(s) of the point, and

$\alpha = [\ \alpha_1\ \alpha_2\ \alpha_3\ ]'$ is a vector of constant coefficients, the 'generalised coordinates'.

In this way, the displacement 'u', which is a scalar here, may be approximated over the region of the element. The generalised coordinates cannot be chosen at random however, since compatibility necessitates that (a) must apply also to the nodes, whose displacements are given as :-

$$\begin{aligned} u_1 &= \alpha_1 + \alpha_2 z_1 + \alpha_3 z_1^2 \\ u_2 &= \alpha_1 + \alpha_2 z_2 + \alpha_3 z_2^2 \\ u_3 &= \alpha_1 + \alpha_2 z_3 + \alpha_3 z_3^2 \end{aligned} \quad \text{or} \quad \mathbf{q} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \Phi \alpha \quad \text{where} \quad \Phi = \begin{bmatrix} 1 & z_1 & z_1^2 \\ 1 & z_2 & z_2^2 \\ 1 & z_3 & z_3^2 \end{bmatrix}$$

It follows that, for compatibility, the generalised coordinates must satisfy :-

$\alpha = \Phi^{-1} \mathbf{q}$ so, from (a) $\mathbf{u} = \phi' \alpha = \phi' \Phi^{-1} \mathbf{q} = \mathbf{N q}$ that is $\mathbf{N} = \phi' \Phi^{-1}$

Inserting some values, to show the form of these variables, suppose an element to extend from $z_1 = 4$ to $z_2 = 6$. Therefore $\Phi = \begin{bmatrix} 1 & 4 & 16 \\ 1 & 6 & 36 \\ 1 & 5 & 25 \end{bmatrix}$

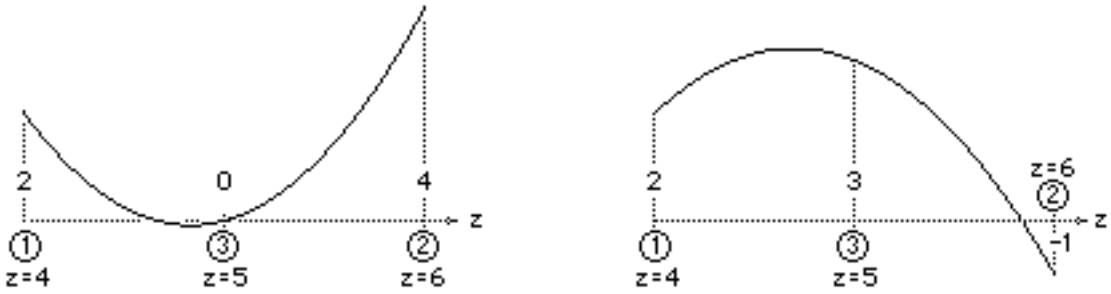
From the above :-

$$\mathbf{N} = \phi' \Phi^{-1} = [\ 1 \quad z \quad z^2\ ] \begin{bmatrix} 15 & 10 & -24 \\ -11/2 & -9/2 & 10 \\ 1/2 & 1/2 & -1 \end{bmatrix} = {}^1/_2 [\ (30-11z+z^2) \quad (20-9z+z^2) \quad (-48+20z-2z^2)\ ]$$

N having thus been found for the element, the continuous variations of displacement, for two possible sets of nodal displacements are as follows :-

$\mathbf{q} = [\ 2\ 4\ 0\ ]'$ $\qquad$ $\mathbf{q} = [\ 2\ -1\ 3\ ]'$

$\mathbf{u} = \mathbf{N q} = 70 - 29z + 3z^2 \quad (4 \le z \le 6)$ $\qquad$ $\mathbf{u} = \mathbf{N q} = {}^1/_2 (-104 + 47z - 5z^2)$



The form of **N** should be noted particularly; the reason for its name is apparent - it shapes a 'curve' of the chosen order (2nd here) between the nodal displacements, whatever these might happen to be. Note that the number of generalised displacements (three here) must tally with the total nodal degrees of freedom, so that their solution is possible. Thus, in two dimensions (x,y) with two degrees of freedom ($\mathbf{u} = [\ u\ \ v\ ]'$) and say complete second order, we might have :

$u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 x y + \alpha_6 y^2$

$v = \alpha_7 + \alpha_8 x + \alpha_9 y + \alpha_{10} x^2 + \alpha_{11} x y + \alpha_{12} y^2$ eg. six nodes are necessary.

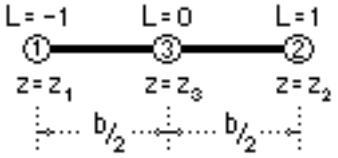
If the plane element for which this model is chosen has less than six external nodes, then the internal nodes may be condensed out using the principle of minimum potential energy (see below for the meaning of condensation). Alternatively, terms may be dropped whilst preserving symmetry of the co-ordinate powers, to tally with the number of external nodes, thus :-

5 nodes : $u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x^2 + \alpha_5 y^2$ or, if there are not 5 nodes,

4 nodes : $u = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 x y$ and similarly for 'v'.

Natural Coordinates

In the one-dimensional element above, the location of a point was defined by global coordinates (z). Natural coordinates on the other hand, specify a point within an element by a set of dimensionless numbers which assume unit value, for example, when the point coincides with an external node. Thus the point is located with respect to the element's nodes, independently of how the element is sized or positioned in the global system. Natural co-ordinates are intrinsic to the element. Thus, in the example above, suppose a natural co-ordinate 'L' is chosen so that L= –1 at node '1' and L= +1 at node '2' as shown, so the linear mapping of 'x' into 'L' is :



$$L = (2z - (z_1 + z_2)) / b \quad \text{or conversely :} \quad z = (b L + (z_1 + z_2)) / 2$$

It follows that L= 0 at the central internal node '3'. Using this mapping, differentiation and integration may be expressed as :-

$$\partial/\partial z = (2/b) \partial/\partial L \qquad \text{and, for integer index 'p', the integral over the length 'b'}$$

$$\int_b L^p \, dl = \int_{z_1}^{z_2} L^p \, dz = {}^{b}/_{2} \int_{-1}^{+1} L^p \, dL = {}^{b}/_{2} \, L^{p+1}/(p+1) \div_{-1}^{+1}$$

$$= 0 \text{ if p is odd, or } = {}^{b}/_{(p+1)} \text{ if p is even - a very simple result.}$$

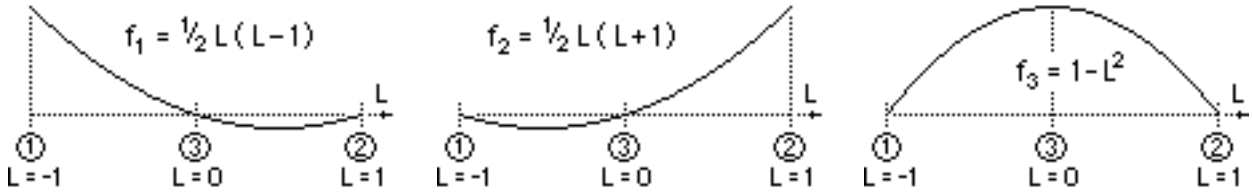
We now set up, corresponding to each node, an 'interpolation function, f', which assumes a value of unity at the node, and zero at other nodes. Thus, by inspection :-

'1' : $f_1 = {}^{1}/_{2} \, L (L - 1)$ ie $f_1 = 1$ @ '1' when L = -1 and $f_1 = 0$ @ '2' & '3'

'2' : $f_2 = {}^{1}/_{2} \, L (L + 1)$ ie $f_2 = 1$ @ '2' when L = +1 and $f_2 = 0$ @ '3' & '1'

'3' : $f_3 = (1 - L^2)$ ie $f_3 = 1$ @ '3' when L = 0 and $f_3 = 0$ @ '1' & '2'

These three interpolation functions are plotted below :



We thus have, in general for this element :

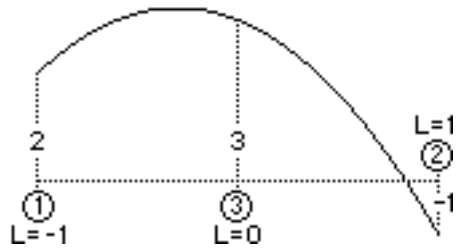
$$z = [f_1 \; f_2 \; f_3] [z_1 \; z_2 \; z_3]' = f_1 z_1 + f_2 z_2 + f_3 z_3$$

$$= [{}^{1}/_{2} L(L-1) \; {}^{1}/_{2} L(L+1) \; (1 - L^2)] [z_1 \; z_2 \; z_3]'$$

We can then use the same shape factor and say immediately that :

$$u = [{}^{1}/_{2} L(L-1) \; {}^{1}/_{2} L(L+1) \; (1 - L^2)] [u_1 \; u_2 \; u_3]' \qquad \text{eg } u = u_1 \text{ when L = -1 \&c}$$

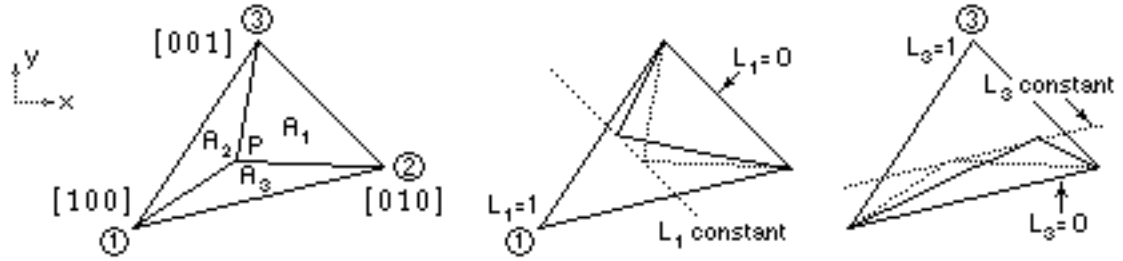
This is just the equation : $\mathbf{u} = \mathbf{N q}$ again, however **N** is now expressed in terms of the natural coordinate, L, rather than as previously through the global Cartesian z. By way of illustration, suppose that the vector of nodal displacements here happens to be $\mathbf{q} = [2 \; -1 \; 3]'$. Inserting this into the above yields $u = (6 - 3L - 5L^2) / 2$, which plots out exactly as before :-



Elements which employ the identical shape function to define the location and displacement of the general point are termed 'isoparametric' elements.

When dealing with triangular elements, the most straightforward natural coordinates with which to define the general point 'P', are the area ratios 'L_i' sketched below; only two of these are independent :-

$$\mathbf{L} = [L_1 \; L_2 \; L_3]' \qquad \text{where } L_i = A_i / A \;\; (0 \le L_i \le 1); \;\; A = \Sigma A_i ; \;\; i = 1, 2, 3$$



The mapping of the global and natural coordinates may be written :-

$$\begin{bmatrix} 1 \\ x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} \qquad \text{eg} \quad L_1 + L_2 + L_3 = 1 \quad \text{and}$$

$$x = x_1L_1 + x_2L_2 + x_3L_3 \quad \text{or, conversely :-}$$

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \end{bmatrix} = (1/2A) \begin{bmatrix} 2A_{23} & b_1 & a_1 \\ 2A_{31} & b_2 & a_2 \\ 2A_{12} & b_3 & a_3 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \end{bmatrix}$$

where $a_i = - x_j + x_k$; $b_i = y_j - y_k$ and $2A_{ij} = x_i y_j - x_j y_i$ with cyclic permutation.

Differentiation and integration, analagous to the one-dimensional equations, are

$$\partial/\partial x = \Sigma_{i=1}^{3} (\partial L_i/\partial x)\, \partial/\partial L_i = (1/2A) \Sigma_{i=1}^{3} b_i\, \partial/\partial L_i \; ; \quad \partial/\partial y = (1/2A) \Sigma_{i=1}^{3} a_i\, \partial/\partial L_i$$

$$\int_A L_1^{\,p}\, L_2^{\,q}\, L_3^{\,r}\, dA = 2A\, p!\, q!\, r! \,/\, (p+q+r+2)! \qquad \text{again, a simple result.}$$

The Strain-Displacement Matrix for the Constant Strain Triangle

Using triangular natural coordinates, the **B** matrix for the constant strain triangle is as follows.

In two dimensions, the kinematic relationship between strain $[\, \varepsilon_x \; \varepsilon_y \; \gamma_{xy} \,]'$ at a point $[\, x \; y \,]'$, and the displacement $[\, u \; v \,]'$ at the point is :-

$$\begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} [\, u \;\; v \,]'$$

that is the equation : $\varepsilon = \partial\, \mathbf{u}$ above, in complete detail.

For constant strain, a linear displacement model is appropriate - ie a point's displacement components are directly proportional to its coordinates. Using the isoparametric concept in conjunction with the triangular natural coordinates above, we have immediately that :-

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} L_1 & L_2 & L_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_1 & L_2 & L_3 \end{bmatrix} [\, u_1 \;\; u_2 \;\; u_3 \;\; v_1 \;\; v_2 \;\; v_3 \,]' \quad \text{ie} \quad \mathbf{u} = \mathbf{N}\, \mathbf{q}$$

From the above :-

$$\varepsilon_x = \partial u/\partial x = \partial/\partial x\, (\, L_1 u_1 + L_2 u_2 + L_3 u_3 \,) \qquad \text{and so, using the chain rule}$$

$$= (\, b_1 u_1 + b_2 u_2 + b_3 u_3 \,) \,/\, 2A$$

Carrying out similar steps for the other strain components leads to :-

$$(\mathbf{b}) \quad \begin{bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{bmatrix} = 1/2A \begin{bmatrix} b_1 & b_2 & b_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & a_2 & a_3 \\ a_1 & a_2 & a_3 & b_1 & b_2 & b_3 \end{bmatrix} \mathbf{q} \qquad \text{that is,} \quad \varepsilon = \mathbf{B}\, \mathbf{q}$$

The strain-nodal displacement matrix, **B**, can thus be evaluated in terms of the element's nodal coordinates which are embodied in the 'a_i' and 'b_i' terms. **B**, and the strains are constant over this element (of thickness 'h') so integration for the element's stiffness is trivial :

$$\mathbf{k} = \int \mathbf{B}'\, \mathbf{C}\, \mathbf{B}\, dV = hA\, \mathbf{B}'\, \mathbf{C}\, \mathbf{B}.$$

Finite Elements - Implementation

We have seen that, overall, application of the FEM requires the following steps :

- Discretise the continuum (define the topology) into a network of membrane elements, or axi-symmetric thick shell elements - or whatever element is appropriate to the body's shape
- Select the displacement model (the order of the virtual displacement-coordinate equation) and hence the complexity of the elements' constitutive laws
- Derive the element stiffness matrices, as exemplified above
- Assemble the equations, using direct assembly for example, and substitute values for the specified displacements and loads
- Solve the equations for the vector of nodal displacements, **q**
- Compute element stresses and strains from the constitutive laws.

We shall refer to the elementary demonstration package 'FEM1' which is typical in that the user is required to carry out the first step only. The package is restricted to two-dimensional linear elastic problems; its basic elements are quadrilateral and triangular laminae with a node at each corner. These are subdivided by the program into constant strain triangles whose **B**-matrix has been derived in (b) above.

The user communicates with the package via a data file. The appended 'FEM1 User's Guide' should be consulted for full details of file preparation, but essentially the prototype is first subdivided into quadrilaterals (and/or triangles) by the user, who then enters the coordinates of the resulting nodes into the data file, together with the node indices which define each element. Element thickness, material properties and prescribed loads and displacements are also entered into the file.

The program reads the data file element by element, reading each element's geometry, then evaluating its stiffness matrix from this data, and finally assembling the submatrices directly into the system equations before moving on to the next element. Which is all very similar to the linear 1-net procedure above.

Before the discretisation process itself is examined, it is necessary to consider a couple of topics which are relevant to it.

Condensation and Bandwidth

Condensation is an elimination process applied during the solution of a set of simultaneous equations, in which a selected variable is eliminated between one particular equation and the others, the coefficients of that equation being stored temporarily out of core, on disc for example. This is exemplified for x_4 :-

Original set of simultaneous equations :-

$$\begin{aligned} \ldots + 3x_3 + 4x_4 - 6x_5 \quad + 2x_7 \quad \ldots. &= 8 \\ \ldots \quad + 2x_4 + 4x_5 + x_6 - 3x_7 \quad \ldots. &= -6 \\ \ldots - 6x_3 - x_4 + 5x_5 \quad - x_8 \quad \ldots. &= 10 \end{aligned}$$

Equation whose coefficients are stored :-

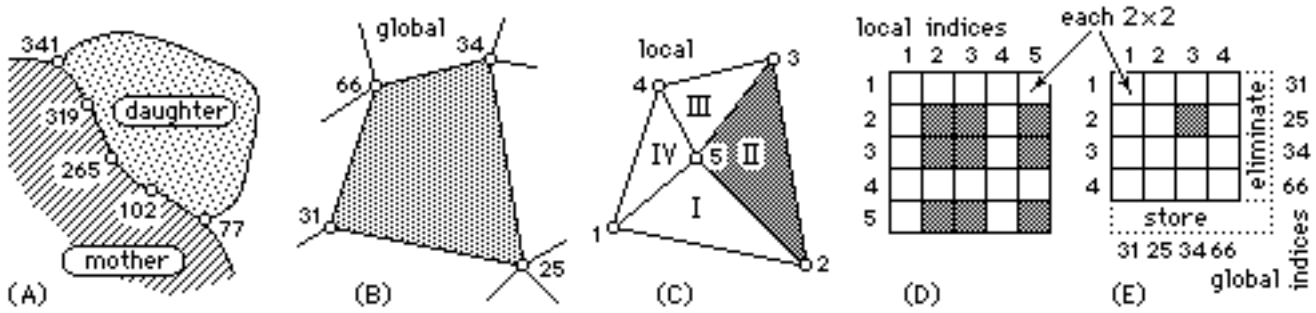
$$x_4 = \ldots - 6x_3 + 5x_5 - x_8 \ldots. - 10$$

Condensed set for solution in core :-

$$\begin{aligned} \ldots -21x_3 + 14x_5 \quad + 2x_7 - 4x_8 \quad \ldots. &= 48 \\ \ldots -12x_3 + 14x_5 + x_6 - 3x_7 - 2x_8 \quad \ldots. &= 14 \end{aligned}$$

The reduced set of equations is then further processed, and eventually solved in core for the variables not so selected. The equation coefficients may later be retrieved and the selected variable evaluated.

A number of variables may be eliminated in this way, thus increasing the size of system which can be handled in a restricted core; for example the daughter sub-system of (A) below is con-



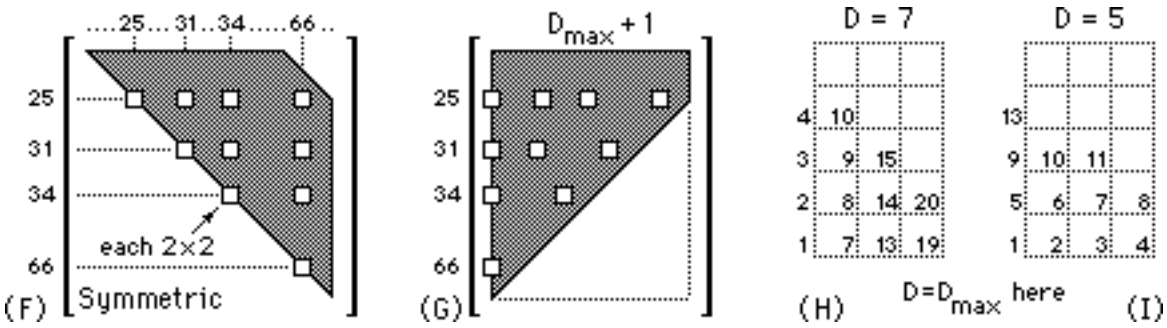
nected to the larger mother only via the five nodes 77-102-265-319-341. The daughter's matrices can be assembled initially as separate entities, then all the displacement variables except $\mathbf{q}_{77}$, $\mathbf{q}_{102}$ etc condensed out. These five remaining equations are then assembled as part of the mother's matrices, which are then solved.

This avoids having to assemble in core the much larger mother-plus-daughter system. The mother might represent the skin of a ship's hull, the various bulkheads being treated as offspring, and so on.

Another application for condensation concerns the basic elements used in FEM1 - the quadrilateral consists of four constant strain triangles. The 2*2 stiffness sub-matrices for a single constant strain triangle are contained algebraically in the source code and can be evaluated once the material and geometric data are supplied - the **B** matrix from equation (b) above, and the **C** matrix corresponding to the element's material.

The triangles' stiffness sub-matrices are assembled after evaluation by the following process, typified by the element defined globally by 31-25-34-66 at (B) above. The program assigns local node indices 1-2-3-4, computes the centroid and defines it as the central node 5 - which is isolated from the rest of the net - thus splitting the element into four triangles (C). The stiffness sub-matrices of each triangle are computed in turn and assembled into the quadrilateral's 5*5 local matrix, as shown at (D) for triangle II. Once this local matrix is complete, the displacement of the central node is condensed out (E), and the resulting 4*4 matrix assembled into the system matrix; eg the shaded sub-matrix is added to the (25, 34) locations of the system matrix. All this is transparent to the user and is aimed at increasing the network's verisimilitude (degrees of freedom), with simple elements and without excessive core demand. The triangular element is similar, being subdivided into three constant strain triangles.

The sub-matrices of the element appear in the system stiffness matrix as shown at (F) below, in which the shaded area has been set aside in core for the problem. Due to symmetry, the upper half only needs be stored, as indicated at (G) in which the dominant diagonal sub-matrices are relocated in the first column of the skewed band.



The number of sub-matrix columns required by any element is (D +1) where D is the difference between the maximum and minimum node indices for the element.

As the size of the sub-matrices equals the number of dimensions, two here, the system as a

whole requires a semi-bandwidth, or number of columns of scalars :

$$B = (D_{max} + 1) * \text{number of degrees of freedom} ;$$

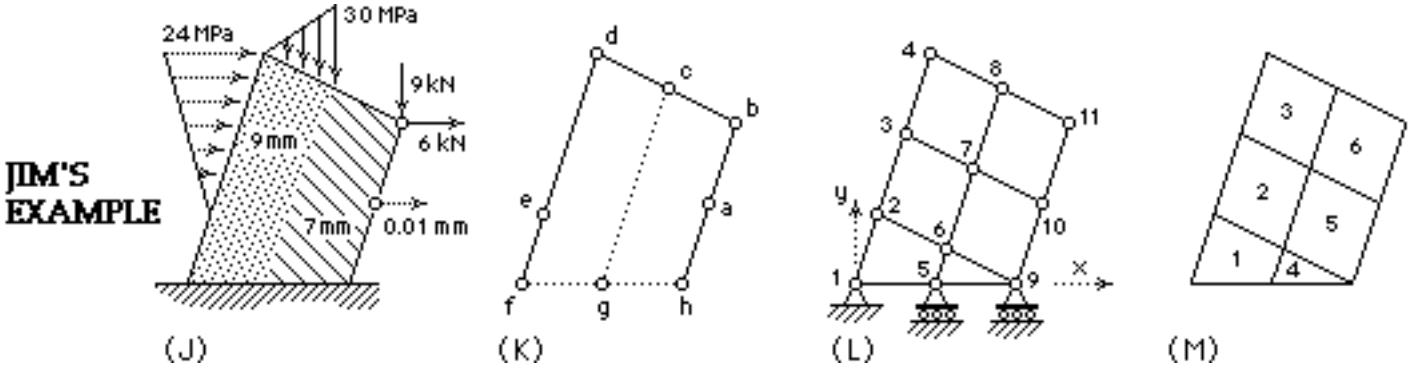
where D_{max} is the maximum difference of elements' nodal indices encountered in the net - and obviously depends upon how the nodes are numbered by the user, as indicated by the nets (H) and (I).

It is necessary to construct and index meshes so that the number of nodes and the semi-bandwidth are not excessive, for although the user of FEM1 can pre-declare the maximum number of rows and columns to cover the demands of any reasonably sized problem, large arrays will lead to inefficient and inaccurate solution. This construction of the mesh requires ingenuity, for at the same time it is usually desirable to incorporate as many elements as possible to obtain an accurate representation of stress variations in way of high stress gradients.

The *frontal method* and *diakoptics* are solution schemes which are not subject to such restrictions, but are rather too complex for our present purposes.

Discretisation

Some further points which should be borne in mind whilst preparing a net will now be exemplified via the prototype cantilever illustrated at (J) below. This consists of an inclined plate of known material and geometry, and two different thicknesses. A concentrated force is applied at one corner, one point on the edge is subjected to a specified displacement, and there are two linearly distributed loads which the program replaces by equivalent concentrated nodal forces.



Nodes must be placed (K) :-

at b, d, f, h	- the discontinuities of the boundary
on line c-g	- line of discontinuity of material property (thickness)
on line f-h	- line of discontinuity between body and supports
at a, b	- points of application of specified loads or displacements
at c, d, e	- end points of distributed loads

Based upon this reasoning the mesh of (L) is drawn up and the nodes indexed. A Cartesian system is set up for the definition of the node coordinates and the senses of the force / displacement components. One support node (#1) is fixed to prevent rigid body motion; other support nodes are arranged on the assumption that, in this particular example, the physical support is stiff only in the vertical direction. The elements are annotated in (M) - there are five quadrilaterals and a single triangle. Element 3 for example, is defined by nodes 8-4-3-7.

This completes the discretisation; there are 6 elements, 11 nodes and the semibandwidth, B, is 12. A data file is then prepared to the format explained in the 'User's Guide', for reading by FEM1. The stresses output by the program will be rather meaningless here because the demonstration mesh is far too coarse.

As a general guide, a mesh should be based upon orthogonality, with elements as well-conditioned as possible to avoid mathematical ill-conditioning and a consequent tendency towards inaccuracy - thus ideally quadrilaterals should be square and triangles equiangular. The mesh should be refined in way of expected stress gradients and concentrations. Some techniques for local refinement are shown here to illustrate the use of triangles to minimise ill-

conditioning. The degree of refinement, and hence the number of nodes and bandwidth, will of course depend upon the problem - the stress gradients which occur, the accuracy sought and the capacity of the implementation. Some of the consequences of refining are dealt with in the Appendix. Full advantage should be taken of any symmetry implicit in the prototype.

Consider half a beam in pure bending, supported at its left end as shown opposite. Instead of this support, another linear traction, equal and opposite to the load on the right, might be applied to the left end. Equilibrium is thus theoreticaly assured - ***BUT*** this may give rise to numerical problems with FEM1, because round-off errors lead to loss of equilibrium, and although net forces might be small, rigid body motion results. So loads and reactions must not be over-specified when defining the model.

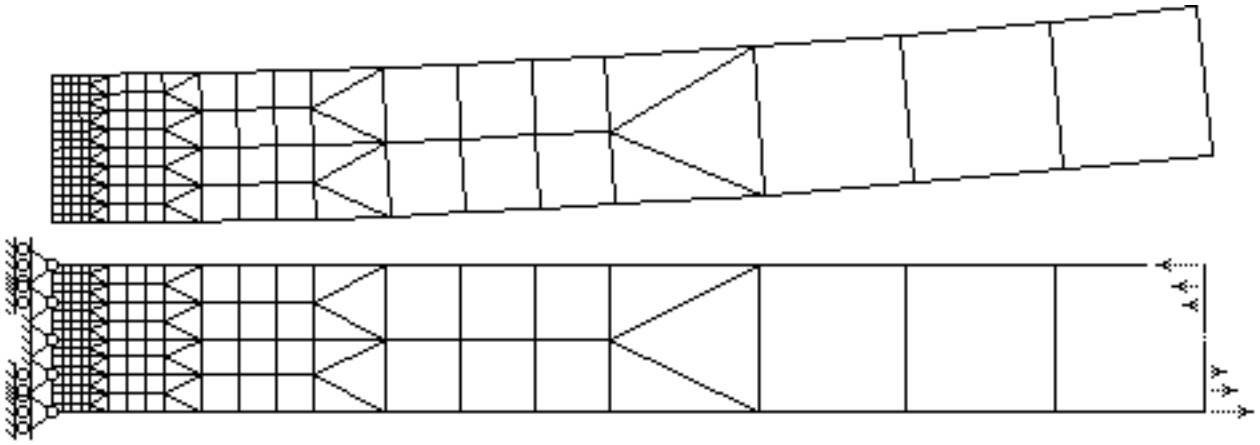
Bibliography

A whole literature exists on this subject; the following are particularly recommended :

Desai CS & Abel JF, **Introduction to the Finite Element Method**, van Nostrand 1972
OC Zienkiewicz, **The Finite Element Method**, McGraw Hill 1977
Irons B & Ahmad S, **Techniques of Finite Elements**, Ellis Horwood 1980

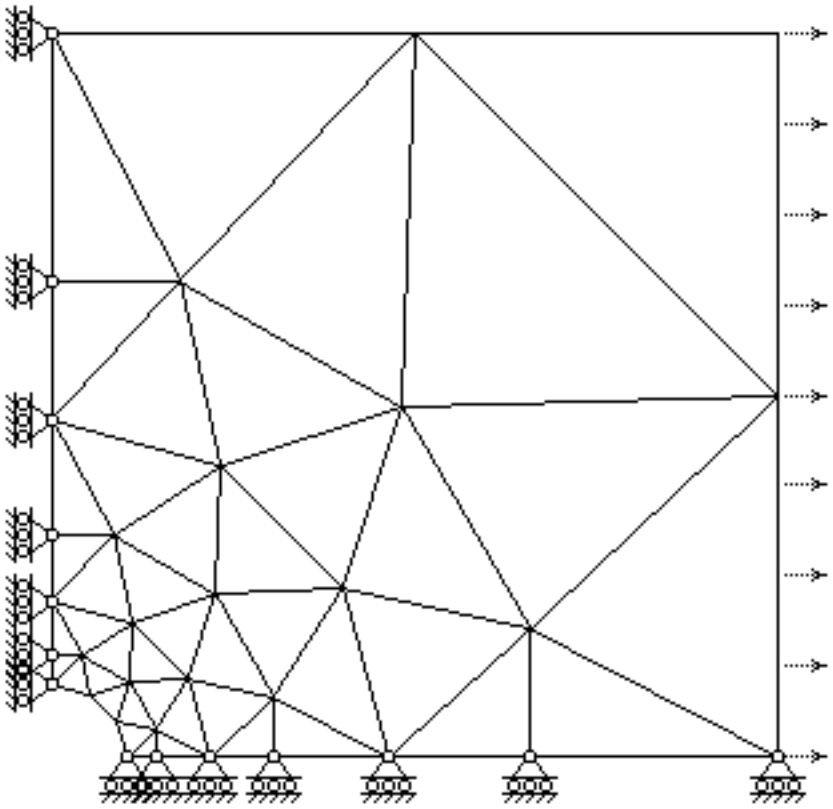
See also :

Kardestuncer H (ed), **Finite Element Handbook**, McGraw-Hill 1987
Barn NM, **Finite Element Analysis on Microcomputers**, McGraw-Hill 1988



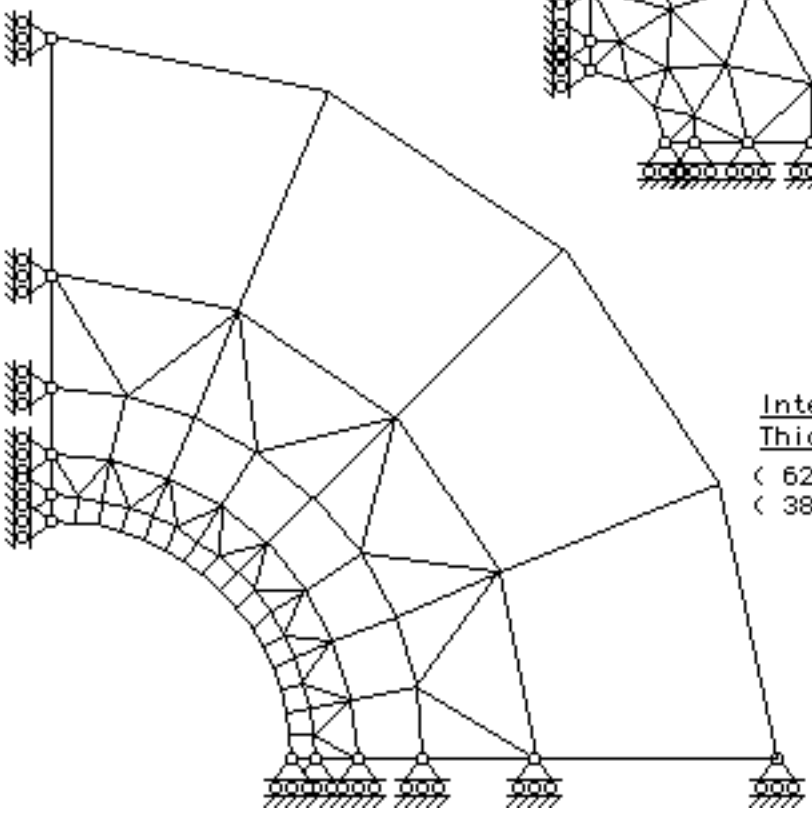
Pure Bending
of a Beam

(161 nodes)
(38 semi-bandwidth)



Stress Concentration
due to a Round Hole
in a Square Plate

(32 nodes)
(14 semi-bandwidth)



Internally Pressurised
Thick Cylinder

(62 nodes)
(38 semi-bandwidth)

FEM1

SOME TYPICAL
DISCRETISATIONS