A finite element/shape interface for the CAD of car structures

A FINITE ELEMENT/SHAPE INTERFACE FOR THE CAD OF CAR STRUCTURES

by

SIK POR PANG

A MASTER's THESIS

Submitted in partial fulfilment of the requirements

for the award of

MASTER OF PHILOSOPHY

of the

LOUGHBOROUGH UNIVERSITY OF TECHNOLOGY

March 1984

Supervisor: Dr A A BALL

ACKNOWLEDGEMENTS

I am indebted to my Supervisor, Dr A A Ball, for his expert help and guidance throughout the duration of my project. I am also grateful to my industrial supervisor, Dr G G Moore, for his valuable assistance during my training at Austin Rover, Cowley.

Further thanks is due to Mr N Dickens for his help with the composition of the thesis, and to Mr D Li for the illustration.

I greatly appreciate the financial support of the SERC case award, and the financial and collaborative assistance from Austin Rover.

Finally, I thank Mrs Pauline Higgs for her accurate and excellent typing of a difficult script.

SYNOPSIS

The thesis describes the general concepts of interfacing the geometrical description of a car body and the internal structure with the corresponding finite element model. The work was carried out on collaboration with the Austin Rover Group (ARG), Cowley, and a three month period was spent with the company to gain understanding of the underlying problem.

In the first half of the thesis, the concepts of CAD/CAM are introduced, and reference is made to their applications in a number of industrial environments. The finite element method (FEM) is also introduced, with an explanation of the fundamental theory and its application to structural problems. The link between the engineering approach to FEM and the mathematical approach is demonstrated by simple examples.

Chapter 4 represents the main achievement of this work and describes the frame work of a solution to the interfacing problems between the CAD geometry and the finite element model. It introduces the concept of a Structural Data Base (SDB) which contains the CAD geometrical description of the car supplemented by structural data including material properties, build information etc. In effect the SDB is a total numerical description of the structure and contains all the necessary information to construct the finite element model. The implementation problems are clearly identified, and particular consideration is given to mesh generation. A discussion and assessment of existing programs is undertaken in chapter 5.

Finally, suggestions for further work are offered, together with some concluding remarks.

CONTENTS

1 INTRODUCTION

1.1 Integrated CAD/CAM

In mechanical engineering design, as indeed in many other branches of design, geometry plays a fundamental role; the shape of an object is crucial whether for functional or aesthetic reasons. The use of computer systems in the engineering industry is widespread for both the design and manufacture of components, because the time of manufacture and design can be reduced. The fields of manufacture and design are invariably separate, but it is a natural step to close the gap, so that the amount of effect required to proceed from design to manufacture is reduced. If designs are produced in a computerised form, this data should be represented to the manufacturing department instead of, or with, the drawings.

Advisory Council for Applied Research and Development [1980] has published a report that aims to understand the meanings of CAD and CAM and to concentrate on the use of computers in the design and manufacture of discrete products. They have produced working definitions such as CAD (Computer-Aided Design), which uses a computer based system to assist in translating a requirement or concept into an engineered design, utilizing a data bank of design principles and information (which may be in the form of drawings) for manufacture. Such designs may include 'simulation' which is the modelling of a design and calculation of performance.

CAM (Computer-Aided Manufacture) is the use of information for CAD and Draughting as a direct input to control manufacturing plant (such as numerically controlled machine tools) or inspection and test equipment.

Integrated CAD/CAM involves the linking of design and manufacturing software via a common data base and could also be extended to encompass marketing, buying, production planning and control activities. The CAD part enables different legal requirements to be easily incorporated into a product reducing the penalty that non-standardisation brings, while the CAM part enables the variants to be more economically manufactured.

Among the benefits of an integrated CAD/CAM system are increased productivity, a significant reduction of errors in communication between design and manufacturing and better cost control. By shortening the time taken in manufacture, capital requirements for both machinery and stock can be decreased. Identical parts can be reproduced at any time if the manufacturing instructions are retained and stocks of spares may thus be reduced. The inspection of manufactured products is also easier, since if the characteristics of a product are defined in the computer, automatic gauging and testing can be employed to check that it is to specification.

The Boeing Aerospace Company [NESS 1981][65] has partially integrated CAD/CAM, because of the potential for improved profits from its use. There was an awareness at top levels of the company that the proper application of computers to design and manufacturing processes could reduce production cost, thereby increasing profits and enhancing the company's competitive position.

Lipchin and Litter [1982] have described the methodology for overall manufacturing performance improvement through the allocation of proper CAD/CAM technology in pay-off areas with managerial, organizational, planning and economic implications. It gave an overall planning procedure for CAD/CAM implementation which consists of the following interrelated steps:-

(1) Determine the proper level and schedule of investment in CAD/CAM.

(2) Tailor the appropriate CAD/CAM technology to the company's needs.

(3) Structure and develop the CAD/CAM organisation and on-going management programme.

They reported the success experienced with companies such as Pratt & Whitney, General Electric and Structural Dynamic Research Corporation, proving that achieving maximum benefits from CAD/CAM requires a company-wide implementation effort accompanied by company-wide planning.

Many companies are not recognising the importance of linking their CAD/CAM software packages. The project in collaboration with the Austin Rover Group is concerned with the linking of surface modelling and finite element analysis.

1.2 CAD in the Austin Rover Group

Emmerson [1976] has described the computer-aided design and the numerical control manufacture in Leyland cars and a number of applications. The traditional methods in producing the scale design of the exterior skin shape from a stylist's full-size clay model into two-dimensional engineering drawing was both time-consuming and tedious. Upon completion of this skin line drawing, information is converted onto the exterior templates for application to the full-size clay model and using these as the basis of the body description. The method was prone to error and many problems were experienced in the drawing office in aligning the templates to a common datum, which was described in detail in Giles [1971].

To overcome the problems described above, various improved techniques have been used for obtaining a satisfactory surface description. The method involved using an automatic measuring machine to record a set of three-dimensional coordinates on punched tapes. This unrefined information is fed into the computer and the structure into a data-base, so that any area may be easily accessed.

The lines are then faired individually using a graphic display manipulation and a large draughting machine, since accurate evaluation of the quality of the lines is not possible by direct inspection on the screen owing to the limitations in size and accuracy. Davy [1972] has explained that the problem can be overcome by displaying a graph of variation in radius of curvature along the length of the curve. Experience is essential to make judgements on the 'smoothness' of the drawing. With these restrictions, the company mainly uses those personnel experienced in design and manufacturing with specialist knowledge of the application, with the assistance from the Computer-Aided Engineering (CAE) Department.

Many of the tests necessary to ensure that a new vehicle satisfies the various safety regulations are destructive in nature, and therefore each new requirement can seriously affect the development lead time and cost. The most severe of these tests are the front, rear, and side impacts. The tests either propel the fully laden vehicle into an immovable barrier or impact the stationary vehicle by a moving barrier. In each case the most important consideration is that the occupants should survive the impact.

A number of simulation models have been developed to cover all the barrier impact tests, and to assess vehicle-to-vehicle impacts as described by Emmerson and Fowler [1974]. Programs have also been developed by the Calspane Corporation in America (described by Bartz [1971]), to simulate the crash victim as a complex three-dimensional model for the analysis of an impact between a victim and a set of vehicle contact surfaces and restraint systems.

Recently, Austin Rover have implemented a new interactive computer package CATIA to produce the external and the internal geometrical description of the car bodies. All this data is stored into the Geometrical Data Base (GDB).

1.3 Shape Representation in CAD

We have already remarked that shape is one of the most important variables in engineering design. The accurate representation of shape is correspondingly important to CAD/CAM. There are many mathematical techniques which have been employed in the representation of shape information.

Most CAD systems use the parametric method for shape representation, except, notably, the Autokon system (described by Bates [1972]). Surfaces are presented by vector valued functions of parameters u and v,

$$\overline{P}(u,v) = [x(u,v)\ y(u,v)\ z(u,v)] \quad 0 \leq (u,v) \leq 1.$$

Some of the advantages of parametric representation as pointed out by Sabin [1971] are as follows:-

(1) The vector/tensor notation appropriate can express geometrical relationships very tersely, and that the vector manipulation operations have direct geometrical parallels.

(2) Geometric properties set up in vector terms are usually independent of the coordinate axes used and are thus invariant under rotation and other affine transformations.

(3) The computation of cutter offsets and similar related curves for numerical control purposes can be much simpler, i.e. offsets of skin thickness or machinery cutter compensation can be handly exactly.

(4) Points on curves or surfaces are readily computed sequentially along the curve or along parametrics lines in the surface for display purpose.

Forrest [1972] has classified four basic ways in which surfaces can be constructed as a bivariate function of either zero-variate or univariate data or some mixture of both as shown in Figure (1.3.1)

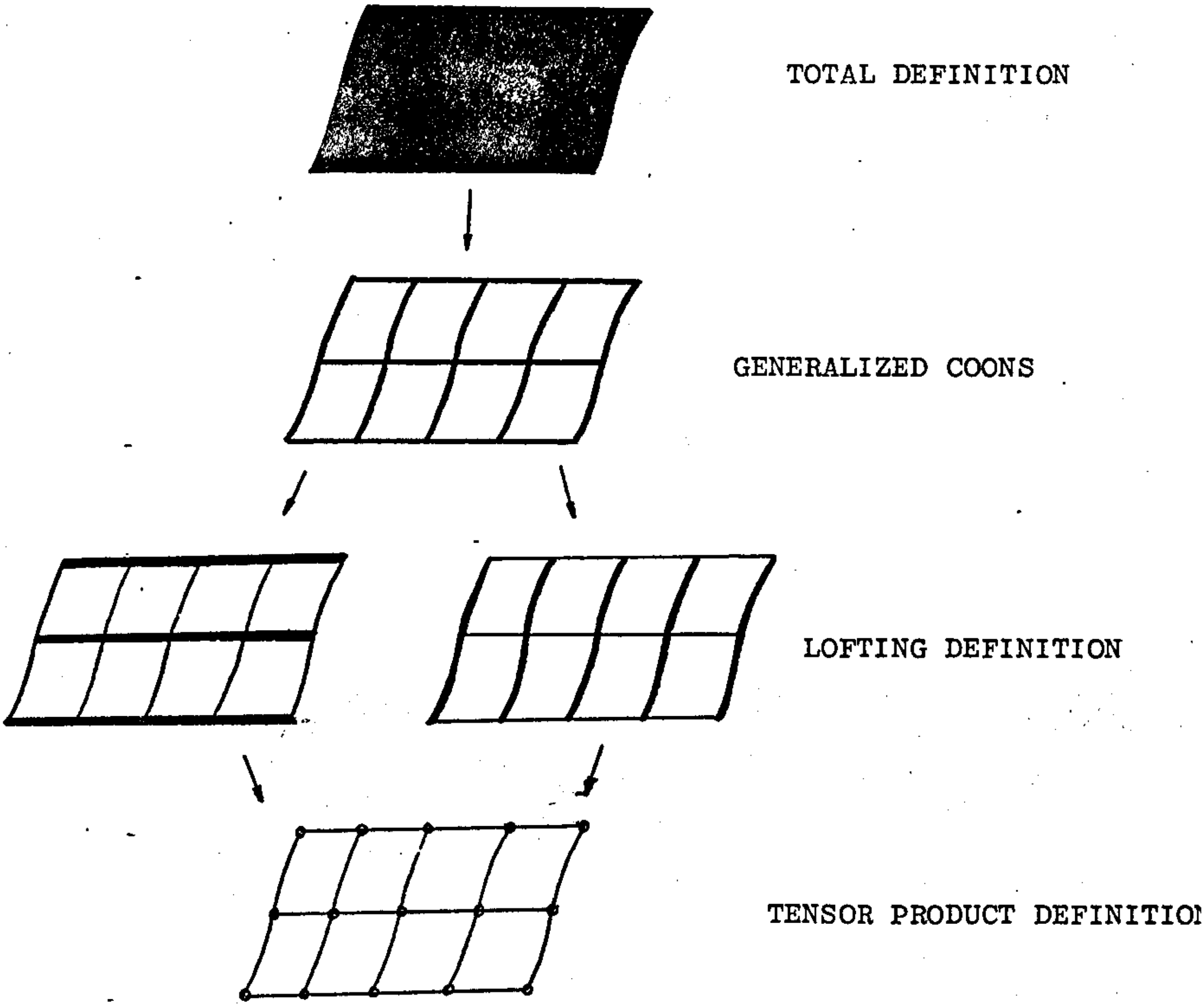


Figure (1.3.1). <u>Sequence of surface definition techniques</u>

At the top of the sequence comes the total definition of surface e.g. by specifying that it shall be a cylinder or a hemisphere. Then comes the generalized Coons method, Coons [1967], Forrest [1970], and Gordon [1971], which formulates the surface in terms of patches defined by their boundary functions. That is to say that four boundary curves of a surface patch are defined, i.e. in terms of two compatible families of univariate functions i.e. curves.

The surface patches are then asembled to form larger surfaces. An example is POLYSURF, Flutter & Rolph [1976].

Next we have Lofting, which corresponds most closely to the manual process, Liming [1944]. It is usually applied to parallel plane cross-sections at a number of longitudinal stations. Examples of the approach are the method of CONSURF, Ball [1974, 1975, 1977] and DUCT, Welbourn [1981].

The fourth method is the tensor-product definition. The method basically defines a surface in terms of zero-variate data, i.e. point vectors or derivative vectors. This is probably the most popular approach. Some notable examples are F-surfaces,Ferguson [1964], NMG,Sabin [1972] and UNISURF, Bezier [1968].

1.4 Finite Element Analysis of Structures

Finite element method is a numerical analysis technique for obtaining approximate solution to problems such as heat flow, solid mechanics, fluid flow, magnetic field calculations, etc. The need for numerical methods arises from the fact that for most practical engineering problems, analytical solutions do not exist. To obtain a solution the engineers must make simplifying assumptions, reducing the problem to one that can be solved. Otherwise numerical procedures must be used.

In the finite element method, the region of interest is divided into numerous connected subregions, or elements, within which approximate functions are used to represent the unknown quantity. The physical concept, on which the finite element method is based, has its origins in the theory of structures. The finite element method is a general numerical procedure for the approximate analysis of arbitrary structures and structural systems. During its original development in the aircraft industry, it was considered merely as a generalization of the well-known displacement method of structural analysis, which has been used extensively for the analysis of frame structures such as bridges, ships hulls and aircraft fuselages.

Since the finite element method is one of the most powerful methods for the approximate solution, it has attracted the attention of mathematicians to establish the fundamental theory. It is, in essence, a variational approximation, employing the Rayleigh-Ritz-Galerkin approach. A given region is represented as a collection of a number of geometrically simpler regions (finite elements) connected together at nodal point. Simple mathematical functions, generally polynomials, are chosen for each element, and the solution over the entire region is obtained by fitting together the individual elements. The isoparametric 'quadratic' element described by Zienkiewicz [1971] is probably the most common type element models.

There are other interpolating methods for meshing structures, such as (1) the Laplacian method, which is described in Buell & Bush [1973] and Herrmann [1971] and (2) the transfinite mapping methods developed by Gordon and Hall [1973].

In structural analysis, the unknown field variables of displacements or stresses are defined in terms of values at the node points, which are the unknowns of the problem. However, the accuracy of the solution depend not only on the number and the size of the finite elements used but also on the interpolation functions selected for the elements.

Using these functions, a number of approaches can be used to formulate the element properties (stiffness and stress matrices). The first of these is the 'direct approach', where physical reasoning is used to establish the element equations in matrix form, which are then combined to form the governing equations for the entire problem. The other approach is more advanced and versatile in the application of the method when extended to other fields. Here, the calculus of variations is used to derive various energy principles which are then employed in deriving the element formulation.

In structural mechanics problems, the element functions are usually chosen to represent displacements within the element (commonly called the displacement method). They could also be chosen to represent stresses (force or equilibrium method), or a combination of displacements and stresses (the hybrid method), which was introduced by Pian [1964][51]. For most problems, the displacement method is the simplest to apply; and is consequently the most widely used.

Usually, a structure such as a car body is manually inputted or digitized into the computer for the construction of finite element models, but these techniques only provide a skeleton model. However, we can now have an accurate geometrical description of the structure using the facilities of CAD.

An automatic mesh generation program will fulfill the task of meshing the complex structures as well as overcoming all the difficulties mentioned above and minimising the errors caused by the user. However, so far, there is no such program that can fully mesh complex structures automatically without user interruptions during the process, (see Chapter 4 for details).

The finite element procedure is often implemented with large general-purpose programs, perhaps the most widely used at this time is the NASTRAN program developed by NASA, which is described by MacNeal [1976].

1.5 Finite Element Method Used In Automobile Design

Before the development of numerical techniques, the complexity of automobile components made analytical prediction of structural behaviours difficult, and in many cases, even impossible. Automobile design, therefore, from the beginning of the industry, relied mainly on the test results and field data evaluations. Although testing has resulted in many new design innovations and a number of improved products, it remains a time-consuming and costly process. The requirement for improved performance in vehicle design has led to increased complexity and costs.

Finite element techniques can fulfil the tasks of reducing costs as well as saving time-[illegible]. The method has already been widely used in many automobile companies such as Ford, British Leyland, etc. The application of the finite element analysis can reduce the number of prototype parts needed to be built and tested. The finite element method is combined with computer facilities to analyse the behaviour of the car body. The types of analysis carried out are usually static testings, which is mainly concerned with the response of a structure to single load input, and dynamic testings, which is concerned with the physical or audio responses of a structure to an infinite number of inputs.

1.6 CAD/FEM Interface

CAD and FEM are separate established technologies. There is an increasing requirement to link the software in these two technologies. Butlin [1983] reports that there is no company providing a whole system which includes a comprehensive CAD system as well as a comprehensive FEM system. Consequently, passing data between these systems is, at least, a loose linkage and, at worst, an uncomfortable, tedious and error-prone process.

However, it would be a considerable advantage to be able to use a CAD system in the data generation for FEM. In particular it would enable the physical shapes of the complex structure to be accurately represented mathematically in the computer.

The aim of this project is to write an automatic mesh generation program that can use the geometrical data of a car body, which is created in Austin Rover, using CATIA and CADAM. The FEM mesh generator must therefore be capable of receiving and utilising this geometry.

2 APPROACHES TO FINITE ELEMENT ANALYSIS

2.1 History of The Finite Element Method With an Outline of Chapter Contents

The finite element method first appeared in the 1950's as a technique for handling structural engineering problems. It was an outgrowth of the so-called matrix method of structural analysis. For about ten years, there were two competing matrix methods, one with the unknown boundary or nodal forces as variables, which was known as the flexibility (force) method, and the other based on unknown displacements as variables, known as the displacement (stiffness) method. These two methods are described in section 2.2. Much of the theory was introduced by Argyris [1960] in a series of papers on energy theorems and matrix methods. In 1956, Turner et al, published a paper on the stiffness method for aircraft structures. The book by Premieniecki [1968] presented the matrix methods as applied to the solution of stress analysis problems.

The name 'finite element method' was coined by Clough [1960] in a paper describing applications in plane elasticity. He showed that the method is based on the principle of minimum total potential energy in terms of prescribed displacement field (see section 2.3.1). The process is equivalent to the Rayleigh-Ritz method.

Although originally derived for structural problems, finite element methods have expanded into such fields as heat transfer, fluid flows etc. The book by Zienkiewicz & Cheung [1967] presented the broad application of the method. Later, Szabo & Lee [1969] showed that the element equations could also be derived by using a weighted residual method such as Galerkin's or least-squares (see section 2.3.2). This led to a widespread interest among applied mathematicians in applying the finite element method for the solution of linear and non-linear differential equations. With this progress, today the finite element method is a well established and convenient analysis tool used by engineers and applied scientists.

2.2 Structural Analysis by Matrix Methods

In the matrix methods of structural analysis, there are two basic approaches used. These approaches are known as the displacement (stiffness) method and the flexibility method. These methods correspond to alternative forms of energy principles, and we shall find it advantageous to develop the methods from an energy standpoint (see section 2.3).

In the displacement method, the displacements of the nodes are unknown and are determined by the equations of equilibrium. It is probably best to illustrate the method by a simple example. Consider a cantilever beam loaded as shown in Figure 2.2.1.

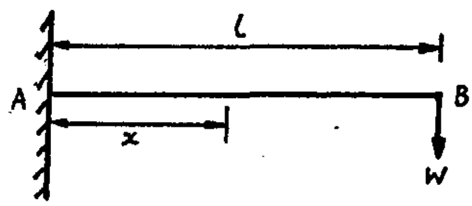


Figure 2.2.1

We use the slope-deflection equations and the notation shown in Figure 2.2.2.

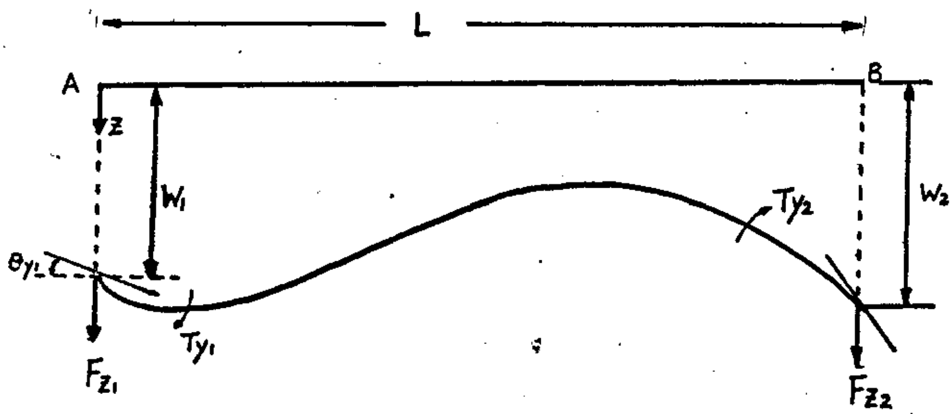


Figure 2.2.2

Let M be the bending moment about B.

Take

$$M = F_{z1}x - T_{y1}$$

Then, comparing with the differential equation of flexure.

$$M = EI\,\frac{d^2y}{dx^2}$$

where I is the moment of inertia and E is the Young's modulus. Then

$$EI\,\frac{d^2y}{dx^2} = F_{z1}x - T_{y1}$$

integrating once,

$$EI\,\frac{dy}{dx} = \frac{1}{2}\,F_{z1}\,x^2 - T_{y1}x + C \qquad (2.2.3)$$

at $x = 0$, $\frac{dy}{dx} = \theta_{y1}$, and therefore

$$C = EI\ \theta_{y1}$$

at $x = L$, $\frac{dy}{dx} = \theta_{y2}$

$$EI\ \theta_{y2} = \frac{1}{2} F_{z1} L^2 - T_{y1} L + EI\ \theta_{y1}$$

$$F_{z1} = 2\frac{T_{y1}}{L} + \frac{2\ EI\ \theta_{y2}}{L^2} - \frac{2\ EI\ \theta_{y1}}{L^2} \qquad (2.2.4)$$

From 2.2.3 and 2.2.4, we have

$$EI\frac{dy}{dx} = T_{y1}\left(\frac{x^2}{L} - x\right) + EI\ \theta_{y1}\left(1 - \frac{x^2}{L^2}\right) + EI\ \theta_{y2}\frac{x^2}{L^2}$$

Integrating again

$$EIy = T_{y1}\left(\frac{x^3}{3L} - \frac{x^2}{2}\right) + EI\ \theta_{y1}\left(x - \frac{x^3}{3L^2}\right) + EI\ \theta_{y2}\frac{x^3}{3L^2} + C_2$$

At $x = L$, $y = w_2$

$$EI\ w_2 = -\frac{T_{y1}\ L^2}{6} + \frac{2EIL\theta_{y1}}{3} + \frac{1}{3} LEI\theta_{y2} + EIw_1$$

$$\text{or} \quad T_{y1} = \frac{4EI\theta_{y1}}{L} + \frac{2EI\theta_{y2}}{L} + \frac{6EIw_1}{L^2} - \frac{6EIw_2}{L^2}$$

$$\text{similar } T_{y2} = \left(\frac{6EI}{L^2}\right) w_1 + \left(\frac{2EI}{L}\right)\theta_{y1} - \left(\frac{6EI}{L^2}\right) w_2 + \left(\frac{4EI}{L}\right)\theta_{y2}$$

For equilibrium, Σm about $B = 0$

$$T_{y1} - F_{z1} L + T_{y2} = 0$$

$$F_{z1} = \frac{T_{y1}+T_{y2}}{L} = -F_{z2}$$

$$= -\left(\frac{12EI}{L^3}\right) w_1 - \left(\frac{6EI}{L^2}\right)\theta_{y1} + \left(\frac{12EI}{L^3}\right) w_2 - \left(\frac{6EI}{L^2}\right)\theta_{y2}$$

These equations may be written in matrix form as follows:

$$\begin{Bmatrix} F_{z1} \\ T_{y1} \\ F_{z2} \\ T_{y2} \end{Bmatrix} = \frac{EI}{L^3} \begin{bmatrix} 12 & & & \\ 6L & 4L^2 & \text{Symmetric} & \\ -12 & -6L & 12 & \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{Bmatrix} w_1 \\ \theta_{y1} \\ w_2 \\ \theta_{y2} \end{Bmatrix} \quad (2.2.5)$$

and summarised as

$$\{F^e\} = [K^e]\{\delta^e\}$$

Where $[K^e]$ is the stiffness matrix, $\{\delta^e\}$ is the vector of displacements and $\{F^e\}$ is the vector of forces.

In the flexibility method, the forces are the unknown quantities and are determined by the equations of equilibrium relating the internal forces in the elements, the external applied forces at the nodes and/or the unknown reactions at the supports. Basically, the flexibility method is dual to the displacement method as shown by the force-displacement relationship.

To illustrate the method, we consider the cantilever beam with two loadings as shown in Figure (2.2.6) and use the notation in Figure (2.2.7).

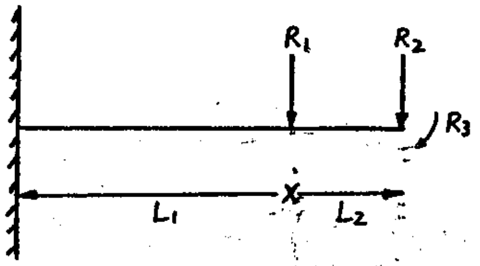


Figure (2.2.6)

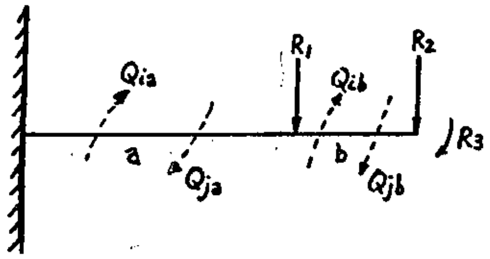


Figure (2.2.7)

$\underline{Q}$, the total set of member force

$\underline{R}$, the total applied load

$\underline{b}$, the force transformation matrix

$$\underline{Q} = \underline{b}\,\underline{R}$$

$$\begin{Bmatrix} Q_i^a \\ Q_j^a \\ Q_i^b \\ Q_j^b \end{Bmatrix} = \begin{array}{c} \begin{array}{ccc} R_1 & R_2 & R_3 \end{array} \\ \begin{bmatrix} -L_1 & -(L_1+L_2) & -1 \\ 0 & L_2 & 1 \\ 0 & -L_2 & -1 \\ 0 & 0 & 1 \end{bmatrix} \end{array} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix}$$

the individual member flexibility matrices are

$$f^a = \frac{1}{6EI}\begin{bmatrix} 2L_1 & -L_1 \\ -L_1 & 2L_1 \end{bmatrix}, \quad f^b = \frac{1}{6EI}\begin{bmatrix} 2L_2 & -L_2 \\ -L_2 & 2L_2 \end{bmatrix}$$

$$f = \frac{1}{6EI}\begin{bmatrix} 2L_1 & -L_1 & & \\ -L_1 & 2L_1 & \multicolumn{2}{c}{0} \\ & 0 & 2L_2 & -L_2 \\ & & -L_2 & 2L_2 \end{bmatrix}$$

$$F = b^T f b$$

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix} = \begin{bmatrix} \frac{L_1{}^3}{3EI} & \frac{2L_1^3 + 3L_1{}^2L_2}{6EI} & \frac{L_1{}^2}{2EI} \\ \frac{2L_1{}^3 + 3L_1{}^2L_2}{6EI} & \frac{(L_1+L_2)^3}{3EI} & \frac{(L_1+L_2)^2}{2EI} \\ \frac{L_1{}^2}{2EI} & \frac{(L_1+L_2)^2}{2EI} & \frac{L_1+L_2}{EI} \end{bmatrix} \begin{bmatrix} R_1 \\ R_2 \\ R_3 \end{bmatrix} \qquad (2.2.8)$$

In general, the flexibility method is not as convenient for computing as the displacement method. One of the reasons is that the equations of equilibrium for a flexibility matrix are unsymmetric and less sparsely populated than the displacement matrix. Therefore, it will be necessary to store all the coefficients and this will make very large demands on the storage requirements of a computer. Also, it is much simpler when assembling all the local stiffness matrices to form the global stiffness matrix. In the displacement method, the constrained structure is determined in a very definite way, whereas in the flexibility method, there are many choices for the released structure. Finally, it is much easier to form the displacement transformation matrix than the force transformation matrix, since the effects of displacements are often localized.

For the cantilever problem above, we can form the analytical solution and use it for comparing results with the above methods. First, return to the differential equation of bending,

$$M = EI\,\frac{d^2y}{dx^2}$$

or $$M = EI\nu'' \tag{2.2.9}$$

Then, $$M = -w\ell\left(1-\frac{x}{\ell}\right)$$

or $$EI\nu'' = -w\ell\left(1-\frac{x}{\ell}\right)$$

Integrating once,

$$EI\nu' = -\,w\ell\left(x-\frac{x^2}{2\ell}\right)+A \tag{2.2.10}$$

Where A is a constant of integration.

Integrating again, we have

$$EI\nu = -w\ell\left(\frac{x^2}{2}-\frac{x^3}{6\ell}\right)+Ax+B \qquad (2.2.11)$$

Now, applying all the necessary conditions in (2.2.10) and (2.2.11), where x = 0, the slope v' = 0 (since the beam must be securely built). Hence A = B = 0.

Therefore, for maximum deflection, where $x = \ell$,

$$\nu_{max} = \frac{-w\ell^3}{3EI} \qquad (2.2.12)$$

The equations (2.2.5), (2.2.8) and (2.2.12) will be demonstarted with numerical section (2.4).

2.3 Variational Calculus

2.3.1 Principle of minimum total potential energy

In the previous sections, we have considered the displacement and force methods for structures in the form $\left[K^e\right]\{\delta^e\} = \{F^e\}$. In this section, we will adopt an alternative method. The method is based on the 'variational principle' in which the total potential energy is defined as a functional, π, of the form

$$\pi = \int_\Omega F(\underline{u}, \frac{\partial \underline{u}}{\partial x}, ----)\,\partial\Omega + \int_\Gamma E(\underline{u}, \frac{\partial \underline{u}}{\partial x}, ----)\partial\Gamma \qquad (2.3.1)$$

where Ω is the surface of each element and Γ its part of boundary $\underline{u}$ is the unknown function such that it satisfies a certain equation set Ω and Γ. The function sought may be a scale quantity or may represent a vector of several variable. F and E denote known functions, of a function $\underline{u}$, which make, π, stationary w.r.t. to small changes, i.e. $\delta\pi = 0$. The unknown function $\underline{u}$ can be approximated by the Rayleigh-Ritz method, i.e.

$$\underline{u} \simeq \hat{\underline{u}} = \sum_i N_i \; a_i^e = [N] \; \underline{a}$$

where $[N_i]$ are shape functions denoting functions of position and a_i^e represents a listing of nodal displacements for a particular element. By substituting this approximation into the expression for the functional, π, (2.3.1) and on differentation w.r.t. to $\underline{a}$, a system of equations for the solution of the problem is

$$\frac{\partial\pi}{\partial\underline{a}} \equiv [K] \; \underline{a} + \underline{f} = 0 \qquad (2.3.2)$$

where $[K]$ is a system matrix and $\underline{f}$ is a vector force.

We have so far dealt with the general approach to the approximate solution of the continuum problem. We can now apply the above approach to elasticity problems. Generally, we can express the general form of (2.3.1) as

$$\pi = U + W \qquad (2.3.3)$$

where U is the strain energy of the deformed structure and W is the potential energy of the external force. With reference to general finite element method textbooks for structural problems, viz Zienkiewicz [1979], we can directly obtain the expression for U and W using the following relations:-

$$\{\underline{\varepsilon}\} = [B] \; \{\underline{q}\} \qquad (2.3.4)$$

and

$$\{\underline{\sigma}\} = [D] \; \{\underline{\varepsilon}\} \qquad (2.3.5)$$

where $\{\underline{\varepsilon}\}$ is the strain vector, $[B]$ is the strain matrix, $[D]$ is the elastic matrix $\{\underline{\delta}\}$ is the stress vector and $\{\underline{q}\}$

are the nodal displacements. Using (2.3.4) and (2.3.5), we obtain

$$U = \frac{1}{2}\int_v \{\underline{\varepsilon}\}^T\{\underline{\sigma}\}\ dv = \frac{1}{2}\int_v \{\underline{q}\}^t[B]^t[D][B]\ \{\underline{q}\}\ dv \quad (2.3.6)$$

and $$w = -\int_v \{\underline{Q}\}^T\{\underline{q}\}\ dv \quad (2.3.7)$$

where $\{\underline{Q}\}^T$ is the nodal forces.

Using the equation (2.3.7), we have that

$$\pi = \frac{1}{2}\int \{\underline{q}\}^t[B]^t[D][B]\{\underline{q}\}\ dv - \int_v \{\underline{Q}\}^t\{\underline{q}\}\ dv$$

To minimise the total potential energy, π, we must differentiate w.r.t. to each nodal degree of freedom, and solve the resultant system of equations.

This gives

$$\int_v [B]^t[D][B]\ dv\ \{q\} = \int_v \{\underline{Q}\}^t\{\underline{q}\}\ dv \quad (2.3.8)$$

which can be expressed as

$$\left[K^e\right]\{\underline{q}\} = \{\underline{f}\} \quad (2.3.9)$$

where $\left[K^e\right]$ is the element stiffness matrix.

To illustrate the construction of $\left[K^e\right]$ again we consider the cantilever example shown in Figure 2.2.1. The differential equation of strain energy in bending is

$$U = \frac{EI}{2}\int \left(\frac{\partial^2 v}{\partial x^2}\right)^2 dx \quad (2.3.10)$$

where the equation for the beam shape is assumed of the form

$$V = a_1 + a_2x + a_3x^2 + a_4x^3$$

where a's are arbitrary constants. The boundary conditions are given by

$V = w_1$ and $\frac{dv}{dx} = -\theta_{y1}$ at $x = 0$

$V = w_2$ and $\frac{dv}{dx} = -\theta_{y2}$ at $x = L$

Now we solve for the arbitrary constants a_1 to a_4 as

$a_1 = w_1$, $a_2 = -\theta_{y1}$, $a_3 = 3(w_2-w_1)/L^2+\frac{1}{L}(2\theta_{y_1}+\theta_{y_2})$

and

$a_4 = \left(2(w_1-w_2)-L(\theta_{y1}-\theta_{y2})\right)/L^3$

Substituting for a_1 to a_4 in terms of the end displacements of the element, V can be expressed as

$V = N_1w_1 + N_2\theta_{y1}+N_3w_2+N_4\theta_{y2}$

N_1 and N_4 are the shape functions. By expressing a_1 to a_4 in terms of the w's and θ's, we can obtain all the shape functions N_i's, i=1(1)4,

as

$$N_1 = (1-3x_1^2+2x_1^3)$$

$$N_2 = L(x_1-2x_1^2+x_1^3)$$

$$N_3 = (3x_1^2-2x_1^3)$$

$$N_4 = L(-x_1^2+x_1^3), \quad x_1 = \frac{x}{L} \tag{2.3.11}$$

Therefore,

$$\frac{d^2v}{dx^2} = \frac{d^2N_1}{dx^2_1} w_1 + \frac{d^2N_2}{dx^2_1} \theta_{y1} + \frac{d^2N_3}{dx^2_1} w_2 + \frac{d^2N_4}{dx^2_1} \theta_{y2} \tag{2.3.12}$$

The loss of potential energy of external loads is given by

$$W = F_{z1}\, w_1 + T_{y1}\, \theta_{y1} + T_{y2}\, \theta_{y2} + F_{z2}\, w_2$$

Differentiating U and W partially with respect to w_1, we have

$$\frac{\partial U}{\partial w_1} = EI \int \frac{d^2v}{dx^2} \frac{\partial}{\partial w_1} \left\{\frac{d^2v}{dx^2}\right\} dx = EI \int_0^L \frac{d^2\hat{v}}{dx^2} \frac{d^2N_1}{dx^2} dx \quad (2.3.13)$$

Therefore,

$$F_{z1} = EI \int_0^L \frac{d^2v}{dx^2} \frac{d^2N_1}{dx^2} dx \quad (2.\ 3\ .14)$$

Substituting (2.3.11) and (2.3.12) into (2.3.14), therefore

$$F_{z1} = EI \int_0^L \left(\frac{d^2N_1}{dx^2} w_1 + \frac{d^2N_2}{dx^2} \theta_{y1} + \text{----} + \frac{d^2N_4}{dx^2} \theta_{y2}\right)\left(\frac{d^2N_1}{dx^2}\right) dx \quad (2.3.15)$$

where

$$N_1 = 1 - 3\left(\frac{X}{L}\right)^2 + 2\left(\frac{X}{L}\right)^3$$

$$\frac{dN_1}{dx} = \frac{-6x}{L^2} + \frac{6x^2}{L^3}$$

$$\frac{d^2N_1}{dx^2} = \frac{-6}{L^2} + \frac{12x}{L^3} \quad (2.3.16)$$

Substituting (2.3.16) into (2.3.15), we get the first term as

$$EI \int_0^L \left(\frac{d^2N_1}{dx^2} w_1\right)\left(\frac{d^2N_1}{dx^2}\right) dx = \frac{12}{L^3} EIw_1 \quad (2.3.17)$$

Similarly, partially differentiating U and W with respect to θ_{y1}, W_2 and θ_{y2} leads to the following equations,

$$T_{y1} = EI \int_0^L \frac{d^2v}{dx^2} \frac{d^2N_2}{dx^2} dx$$

$$F_{z2} = EI \int_0^L \frac{d^2v}{dx^2} \frac{d^2N_3}{dx^2} dx$$

$$T_{y2} = EI \int_0^L \frac{d^2v}{dx^2} \frac{d^2N_4}{dx^2} \, dx$$

$$EI\int_0^L \frac{d^2v}{dx^2}\frac{d^2N_i}{dx^2}\,dx = EI\int_0^L \left(\frac{d^2N_1}{dx^2}\frac{d^2N_i}{dx^2} w_1 + \frac{d^2N_2}{dx^2}\frac{d^2N_i}{dx^2}\theta_{y1} + \frac{d^2N_3}{dx^2}\frac{d^2N_i}{dx^2} w_2 \right.$$

$$\left. + \frac{d^2N_4}{dx^2}\frac{d^2N_i}{dx^2}\theta_{y2} \right) dx, \quad i = 1\ (1)\ 4$$

This integral can be evaluated using the shape functions in (2.3.11) and can be shown to be a similar procedure as in (2.3.17). Therefore, we can express the integral

$$\left[K^e\right] = U = EI\int_0^L \frac{d^2v}{dx^2}\frac{d^2N_i}{dx^2}\,dx = EI\left[\frac{12}{L^3} w_1 + \frac{6}{L^2}\theta_{y1} - \frac{12}{L^3} w_2 + \frac{6}{L^2}\theta_{y2}\right] \text{for } i = 1$$

$$= EI\left[\frac{6}{L^2} w_1 + \frac{4}{L}\theta_{y1} - \frac{6}{L^2} w_2 + \frac{2}{L}\theta_{y2}\right] \text{for } i = 2$$

$$= EI\left[\frac{-12}{L^3} w_1 - \frac{6}{L^2}\theta_{y1} + \frac{12}{L^3} w_2 - \frac{6}{L^2}\theta_{y2}\right] \text{for } i = 3$$

$$= EI\left[\frac{6}{L^2} w_1 + \frac{2}{L}\theta_{y1} - \frac{6}{L^2} w_2 + \frac{4}{L}\theta_{y2}\right] \text{for } i = 4$$

This resulting stiffness matrix is the same as the exact one derived in (2.2.5).

2.3.2 Weighted residual (Galerkin's Method)

The finite element method may be also regarded as a form of the Galerkin's Method [1915]. This method constructs an approximate solution to a differential equation, by requiring that the error between the approximate solution and the true solution be orthogonal to the function used in the approximation.

The application of Galerkin's Method yields the equation

$$\int_R N_\beta \; L(\Phi) \quad dR \qquad\qquad \beta = i,\ j,\ k.$$

where Φ is the unknown parameter and is approximated by

$$\Phi \;=\; \left[N_i,\ N_j,\ N_k\right] \ \{\phi\}$$

and L (Φ) is the differential equation governing ϕ. The term is known as the 'weighted value of residual'.

The method can be illustrated by the same example used in the previous section, in which we can start with the differential equation for a bending beam element, i.e.

$$M = EI \frac{d^4v}{dx^4} \;=\; O$$

Let an approximate solution be given by

$$V = N_1\ V_1 + N_2\ \theta_{y1} + N_3\ V_2 + N_4\ \theta_{y2}$$

where the N_i's are as in section (2.3.1). Therefore, we have

$$M = EI \int_O^L \left(\frac{d^4v}{dx^4}\right) \ N_i \quad dx = O \qquad i = 1\ (1)\ 4 \qquad (2.3.18)$$

The above condition refers to the points in the regions and in order to bring in the boundary forces, we use integration by parts. Integrating by parts,

$$\int_O^L \frac{d^4v}{dx^4}\ N_i\ dx = N_i\ \frac{d^3v}{dx^3}\Bigg|_O^L - \int_O^L \frac{d^3v}{dx^3}\,\frac{dN_i}{dx} \quad dx$$

$$\int_O^L \frac{d^3v}{dx^3}\ \frac{dN_i}{dx}\ dx = \frac{dN_i}{dx}\ \frac{d^2v}{dx^2}\Bigg|_O^L - \int_O^L \frac{d^2v}{dx^2}\ \frac{d^2N_i}{dx^2} \quad dx$$

substituting the above expressions into (2.3.18), we have that

$$\int_0^L \left(EI\,\frac{d^4y}{dx^4}\right) N_i\,dx = EI\int_0^L \frac{d^2v}{dx^2}\,\frac{d^2N_i}{dx^2}\,dx$$

$$+\ EI\left(\frac{d^3v}{dx^3}\,N_i - \frac{dN_i}{dx}\,\frac{d^2v}{dx^2}\right)\Bigg|_0^L = 0 \qquad (2.3.19)$$

The end conditions are $\dfrac{d^2v}{dx^2} = \dfrac{d^3v}{dx^3} = 0$ at (O, L)

Therefore, (2.3.19) becomes

$$\left[K^e\right] \equiv M = EI\int_0^L \frac{d^2v}{dx^2}\,\frac{d^2N_i}{dx^2}\,dx \qquad (2.3.20)$$

which is in the same form as in section (2.3.1). This means the stiffness matrix is exactly the same as in section (2.3.1).

This shows that both approaches lead to identical stiffness matrices, so long as the same functions are used. There are other methods, beside these two, such as the least square method, Lagrange multipliers, etc., which can be found in 'The Finite Element Method', Zienkiewicz [1979], 3rd Edition.

2.4 Examples and Results

At Loughborough University, a computer package PAFEC 75 (Program for Automatic Finite Element Calculations) has been implemented. The package is written based on the finite element method and is used for solving structural problems. PAFEC 75 data is inputted in a modular form. Each module begins with a header or 'module card' giving the headings for the columns which form the remainder of the module.

This card is called the 'contents card'.

We can use the PAFEC to find the displacement value of the example beam as shown in figure (2.2.1) with which we can compare the manual calculations from the previous sections. To construct the input data file, we assume a cantilever beam 3 metre long and a loading of 32 kN in the negative y-direction, as shown in figure (2.4.1). It is restrained at node 1, in the x- and y- directions. The listing of the input data file with the computer result is shown in table (2.4.5).

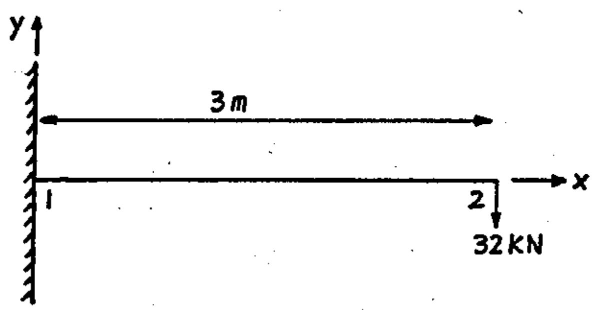


Figure (2.4.1)

The solutions calculated below use the equations formulated in previous sections.

(1) To calculate the analytical solution, we are required to use (2.2.12) and the following conditions:-

A cantilever beam 3m long with cross-section 500mm deep, carries a load of 32kN. Thickness = 4.896×10^{-2} m. $E=209 \times 10^{9} N/m^{2}$.

Therefore, we are required to find the moment of inertia. i.e.

$$I_{zz} = \frac{bd^3}{12} = \frac{(4.896\times10^{-2})(0.5)^3}{12} = (5.1\times10^{-4})\ m^4$$

Torsional constant, $\gamma = 8.644 \times 10^{-7}\ m^4$

Area, $A = 2.448 \times 10^{-2}\ m^2$

Therefore, from (2.2.12), we have

$$V_{max} = -\frac{1}{3}\ \frac{WL^3}{EI} = -\frac{1}{3}\ \frac{(32000)x3^2}{(209x10^9)(5.1x10^{-4})} = \underline{-0.0027019\ m}$$

Bending moment, $m=-wL = -\ (96 \times 10^3\ N\ m)$

Shear force, $Q_x = -w = -32$ kN .

(2) In the displacement method, we are required to use all the equations from which the stiffness matrix in (2.2.5) was derived. From (2.2.5), the stiffness matrix is

$$\overline{K}_{\overline{y}} = \frac{EI_{\overline{y}}}{L} \begin{bmatrix} 12/L^2 & & & \\ 6/L & 4 & & \\ -12/L^2 & -6/L & 12/L^2 & \\ 6/L & 2 & -6/L & 4 \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_{y1} \\ w_2 \\ \theta_{y2} \end{bmatrix}$$

$12I/L^2 = 6.8 \times 10^{-4}$ $\qquad 4I = 2.04 \times 10^{-3}$

$6I/L = 1.02 \times 10^{-3}$

However, since one side is restrained, the stiffness matrix becomes

$$\begin{bmatrix} -32 \times 10^3 \\ 0 \end{bmatrix} = \frac{E}{L} \begin{bmatrix} 6.8 \times 10^{-4} & 1.02 \times 10^{-3} \\ 1.02 \times 10^{-3} & 2.04 \times 10^{-3} \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_{y1} \end{bmatrix}$$

$$\frac{E}{L}\ (w_1(6.8 \times 10^{-4}) + 1.02 \times 10^{-3}\ \theta_{y1}) = -32 \times 10^{-3} \qquad (2.4.2)$$

$$\frac{E}{L}\ ((1.02 \times 10^{-3})\ w_1 + 2.04 \times 10^{-3}\ \theta_{y1}) = 0 \qquad (2.4.3)$$

$\therefore$ (2.4.2) ; (2.4.3)

$\frac{E}{L}\ (3.4 \times 10^{-4}\ w_1) = -64000$

$w_1 = -2.701942 \times 10^{-3}$m.

and $\theta_{y1} = 191999.9985$

$\therefore$ The maximum deflection, $w_1 = \underline{-2.70194 \times 10^{-3}\ m}$

(3) In the flexibility method, we are required to use all the equations used in the derivation of the matrix in (2.2.8). All the boundary conditions and the initial values are as before.

From (2.2.9), we have

$$r_1 = \frac{R_1 L_1^3}{3EI} + R_2 \frac{(2L_1^3 + 3L_1^2 L_2)}{6EI} + \frac{R_3 L_1^2}{2EI}$$

$$r_2 = R_1 \frac{(2L_1^3 + 3L_1^2 L_2)}{6EI} + R_2 \frac{(L_1 + L_2)^3}{3EI} + R_3 \frac{(L_1 + L_2)^2}{2EI}$$

$$r_3 = \frac{R_1 L_1^2}{2EI} + R_2 \frac{(L_1 + L_2)^2}{2EI} + R_3 \frac{(L_1 + L_2)}{EI}$$

When we solve for the values r_i, we have to apply all the necessary conditions, which are as follows:-

$$R_1 = R_3 = 0 \text{ and } R_2 = w$$

Referring to figure (2.2.1),

therefore, we obtain

$$r_2 = R_2 \frac{(L_1 + L_2)^3}{3EI} = \frac{w(L_1 + L_2)^3}{3EI}$$

$$r_2 = \frac{-32 \times 10^3 \times 3^3}{3 \times 209 \times 10^9 \times 3.1 \times 10^{-4}} = \underline{-0.0027019 \text{ m}}$$

which is the resulting vertical deflection of the end of the beam. The resulting rotation of the end of the beam is ,

$$r_3 = R_2 \frac{(L_1 + L_2)^2}{2EI} = \frac{w(L_1 + L_2)^2}{2EI}$$

$$r_3 = \frac{-32 \times 10^3 \times 3^2}{2 \times 209 \times 10^9 \times 5.1 \times 10^{-4}} = \underline{1.35097 \times 10^{-3} \text{m}}$$

From these results, and the computer result in table (2.4.4), where the maximum deflection at node (2)=-0.0027019m we have shown that the modules are correctly defined and the matrices in sections (2.2) and (2.3) are correctly set up.

Table (2.4.4)

```
MAX. BS.LENGTH INCREASED TO   60 BLOCKS.
(    1)  TITLE ANALYSIS BEAM ELEMENT (DBERT)
         ----------------------------------------------------------
----------
(    2)  CONTROL
(    3)  CONCATENATE.OUTPUT
(    4)  STRESS
(    5)  CONTROL.END
(    6)  NODES
(    7)  NODES.NUMBER        X          Y
(    8)  1                  0.0        0.0
(    9)  2                  3.0        0.0
(   10)  3                  3.0        3.0
(   11)  ELEMENTS
(   12)  ELEMENT.TYPE=34000
(   13)  PROPERTIES=1
(   14)  NUMBER           TOPOLOGY
(   15)  1                 1,2
(   16)  BEAMS
(   17)  SECTION.NUMBER=1
(   18)  MATERIAL      IYY       IZZ       TORSIONAL      AREA      NODE.NUMBE
(   19)  1          4.890E-6   5.1E-4    8.0444E-7    2.448E-2       3
(   20)  LOADS
(   21)  CASE.OF.LOAD     NODE.NUMBER    DIRECTION      VALUE.OF.LOAD
(   22)  1                     2              2             -32000
(   23)  RESTRAINTS
(   24)  NODE.NUMBER      DIRECTON
(   25)  1                  12
(   26)  GRAPH
(   27)  FRAME.NUMBER=1
(   28)  TOLERANCE=0.1
(   29)  GRAPH   TYPE    CASE  LIST.OF.NODES
(   30)  1        1        1      1   2
(   31)  IN.DRAW
(   32)  TYPE.NUMBER      INFORMATION.NUMBER
(   33)  2                   23
(   34)  OUT.DRAW
(   35)  PLOT.TYPE     CASE.NUMBER
(   36)  1               1
(   37)  20
(   38)  30
(   39)  END.OF.DATA
0END OF DATA            0 ERRORS
```

CASE 1	TRANSLATIONS			ROTATIONS		
NODE	MULTIPLIED BY 1E 3			MULTIPLIED BY 1E 3		
NUMBER	UX	UY	UZ	PHIX	PHIY	PHIZ
1	*	*	0.0000	0.000	0.000	-1.351
2	0.0000	-2.7019	0.0000	0.000	0.000	-0.000

3 FINITE ELEMENTS AND ASSEMBLY PROCEDURE

3.1 Introduction

In the finite element method, any continium quantity can be approximated by a set of piecewise continuous functions defined over a region. In this Chapter, we construct examples of various types of elements which are used for modelling. We also consider the procedure of assembling the element matrices and vectors to construct the overall system.

3.2 Plane-stress Elements

(I) Plane-stress triangular element

We can obtain the stiffness matrix for the plane-stress triangular element based on the assumed displacement fields and the potential energy integral. The displacements within an element have to be uniquely defined by six values as shown in figure (3.2.2), i.e. two linear polynomials.

$$\begin{aligned} U &= \alpha_1 + \alpha_2 x + \alpha_3 y \\ V &= \alpha_4 + \alpha_5 x + \alpha_6 y \end{aligned} \qquad (3.2.1)$$

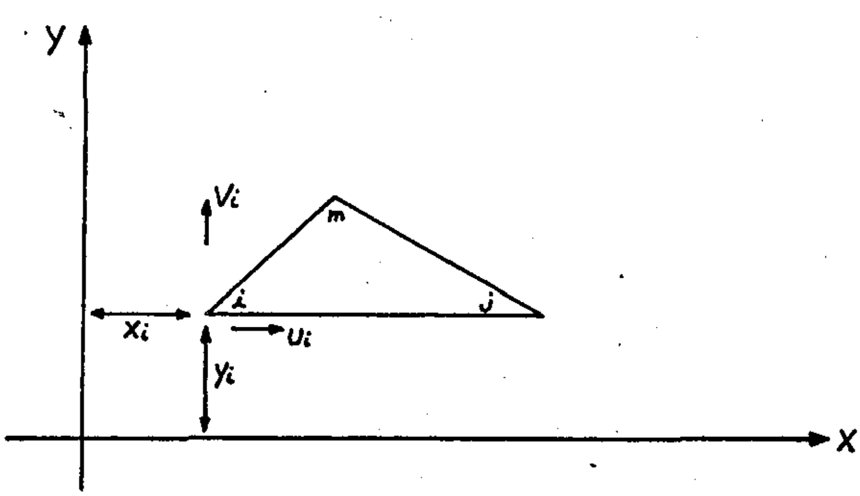


Figure (3.2.2)

We can find the α's in terms of the nodal displacement u's and v's from (3.2.1) and obtain finally

$$U = \frac{1}{2\Delta}\left[(a_i+b_ix+C_iy)U_i+(a_j+b_jx+C_jy)U_j+(a_m+b_mx+C_my)U_m)\right]$$

where the determinant, Δ, is given by

$$\Delta = \frac{1}{2}\begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_m & y_m \end{vmatrix}$$

and $a_i = x_j y_m - x_m y_j$, $b_i = y_j - y_m$, $C_i = x_m - x_j$

Similarly, we also have for v, therefore

$$U = \begin{Bmatrix} u \\ v \end{Bmatrix} = N\underline{a}^e = \left[IN_i \;\; IN_j \;\; IN_m\right] \underline{a}^e$$

and

$$N_i = \frac{1}{2\Delta}\left[a_i+b_ix+C_iy\right] \qquad \text{etc}$$

where N_i are shape functions and $\underline{a}^e$ are nodal displacements. With reference to Zienkiewicz [1979] and the equations shown in section (2.3.1), we can construct the stiffness matrix. Thus, assuming thickness t is constant,

$$K^e = \int B_i^{\,t} D B_j t \; dxdy \qquad i = j = 1, 2, 3$$

where $B_i = \frac{1}{2\Delta}\begin{bmatrix} b_i & 0 \\ 0 & c_i \\ c_i & b_i \end{bmatrix}$ and $D = \frac{E}{(1 - v^2)}\begin{bmatrix} 1 & v & 0 \\ v & 1 & 0 \\ 0 & 0 & \frac{1}{2}(1-v) \end{bmatrix}$

$$\therefore \quad K^e = \frac{Et}{4\Delta^2(1-v^2)}\begin{bmatrix} b_ib_j + \frac{1}{2}c_i(1-v)c_j & b_ivc_j + \frac{1}{2}c_i(1-v)b_j \\ c_ivb_j + \frac{1}{2}b_i(1-v)c_j & c_ic_j + \frac{1}{2}b_i(1-v)b_j \end{bmatrix}$$

II Plane-stress rectangular element

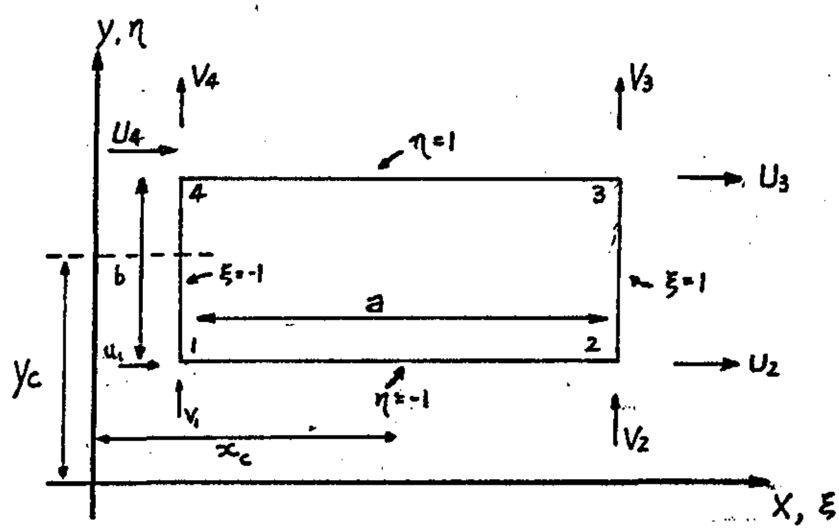


Figure (3.2.3)

In figure (3.2.3), O and C are origins of x, y and ξ, η co-ordinate systems respectively. Conversion from one co-ordinate system to the other is achieved by the relationships

$$\xi = \frac{(x-x_c)}{a} \quad \text{and} \quad \eta = \frac{(y-y_c)}{b}$$

It is convenient to use the same 'origin' for the x, y and ξ, η coordinate systems, thus reducing x_c and y_c to zero become

$$\xi = \frac{x}{a} \quad \text{and} \quad \eta = \frac{y}{b} \qquad (3.2.4)$$

With reference to Zienkiewicz [1979] and the equation (2.3.4) and (2.3.5), where we have the strain matrix, is given by

$$\{\underline{\varepsilon}\} = \begin{Bmatrix} \partial u/\partial x \\ \partial \nu/\partial y \\ \dfrac{\partial u}{\partial y} + \dfrac{\partial \nu}{\partial x} \end{Bmatrix} = [B]\{\underline{q}\} \text{ , where } [B] = \begin{bmatrix} \dfrac{\partial N_i}{\partial x} & 0 \\ 0 & \dfrac{\partial N_i}{\partial y} \\ \dfrac{\partial N_i}{\partial y} & \dfrac{\partial N_i}{\partial x} \end{bmatrix}$$

To obtain the matrix [B], we are required to differentiate the ∂N's expressions with respect to the curvilinear coordinates such as

$$\begin{Bmatrix} \dfrac{\partial N}{\partial \xi} \\ \dfrac{\partial N}{\partial \eta} \end{Bmatrix} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial Y}{\partial \eta} \\ \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \begin{Bmatrix} \dfrac{\partial N}{\partial x} \\ \dfrac{\partial N}{\partial y} \end{Bmatrix} = [J] \begin{Bmatrix} \dfrac{\partial N}{\partial x} \\ \dfrac{\partial N}{\partial y} \end{Bmatrix} \qquad (3.2.5)$$

in which [J] is the Jacobian matrix and can be easily evaluated. We can now obtain $\partial N / \partial x$ and $\partial N/\partial y$ by

$$\begin{Bmatrix} \dfrac{\partial N}{\partial x} \\ \dfrac{\partial N}{\partial y} \end{Bmatrix} = [J]^{-1} \begin{Bmatrix} \dfrac{\partial N}{\partial \xi} \\ \dfrac{\partial N}{\partial y} \end{Bmatrix} \qquad (3.2.6)$$

Therefore, the matrix [B] becomes

$$[B] = \frac{1}{|J|} \begin{bmatrix} \left(a \dfrac{\partial N_i}{\partial \eta} - b \dfrac{\partial N_i}{\partial \xi}\right) & 0 \\ 0 & \left(d \dfrac{\partial N_i}{\partial \eta} - C \dfrac{\partial N_i}{\partial \xi}\right) \\ \left(d \dfrac{\partial N_i}{\partial \eta} - C \dfrac{\partial N}{\partial \xi}\right) & \left(a \dfrac{\partial N_i}{\partial \eta} - b \dfrac{\partial N_i}{\partial \xi}\right) \end{bmatrix}$$

Where $a = \sum_{j=1}^{m} y_j \frac{\partial N_j}{\partial \eta}$, $b = \sum_{i=1}^{m} y_i \frac{\partial N_i}{\partial \xi}$, $c = \sum_{j=1}^{m} x_j \frac{\partial N_j}{\partial \eta}$

and $d = \sum_{i=1}^{m} x_i \frac{\partial N_i}{\partial \xi}$

The stiffness matrix $[K^p]$ becomes

$$\left[K^p\right] = \int_{-1}^{1} \int_{-1}^{1} [B]^T [D] [B] \det |J| d\xi \, d\eta$$

$$[B]^T [D] [B] = \begin{bmatrix} \frac{\partial N_i}{\partial x} & 0 & \frac{\partial N_i}{\partial y} \\ 0 & \frac{\partial N_i}{\partial y} & \frac{\partial N_i}{\partial x} \end{bmatrix} \begin{bmatrix} D_1 & D_2 & 0 \\ D_3 & D_4 & 0 \\ 0 & 0 & D_5 \end{bmatrix} \begin{bmatrix} \frac{\partial N_j}{\partial x} & 0 \\ 0 & \frac{\partial N_j}{\partial y} \\ \frac{\partial N_j}{\partial y} & \frac{\partial N_j}{\partial x} \end{bmatrix}$$

where

$$D_1 = D_4 = \frac{E}{(1-\nu^2)}, \quad D_2 = D_3 = \frac{\nu E}{(1-\nu^2)}, \quad D_5 = \frac{E}{(1-\nu^2)(1-\nu)/2}$$

Therefore,

$$B^T D B = \begin{bmatrix} \frac{\partial N_i}{\partial x} D_1 \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} D_5 \frac{\partial N_j}{\partial y} & \frac{\partial N_i}{\partial x} D_2 \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial x} D_5 \frac{\partial N_j}{\partial y} \\ \frac{\partial N_i}{\partial y} D_5 \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial x} D_2 \frac{\partial N_j}{\partial y} & \frac{\partial N_i}{\partial y} D_4 \frac{\partial N_j}{\partial y} + \frac{\partial N_i}{\partial x} D_5 \frac{\partial N_j}{\partial x} \end{bmatrix}$$

It is difficult to evaluate the stiffness matrix $[K^p]$ without using numerical techniques such as Gauss Quadrature Method, which gives the stiffness matrix $[K^p]$ as follows:-

$$[K^p] = \int_{-1}^{1} \int_{-1}^{1} f(\xi,\eta) \, \partial\xi \, \partial\eta = \sum_{j=1}^{m} \sum_{i=1}^{n} w_i w_j \, f(\xi_i, \eta_j)$$

Where m and n can be equal or different

$$\left[K^p\right] = \sum_{p=1}^{m} \sum_{q=1}^{m} W_p W_q \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix}$$

$$K_{11}= \frac{1}{|J|^2}\left(\left(a\frac{\partial N_i}{\partial \xi}-b\frac{\partial N_i}{\partial \eta}\right)D_1\left(a\frac{\partial N_j}{\partial \xi}-b\frac{\partial N_j}{\partial \eta}\right)+\left(d\frac{\partial N_i}{\partial \eta}-c\frac{\partial N_i}{\partial \xi}\right)D_5\left(d\frac{\partial N_j}{\partial \eta}-c\frac{\partial N_j}{\partial \xi}\right)\right)$$

$$K_{12}=K_{21}=\frac{1}{|J|^2}\left(\left(a\frac{\partial N_i}{\partial \xi}-b\frac{\partial N_i}{\partial \eta}\right)D_2\left(d\frac{\partial N_j}{\partial \eta}-c\frac{\partial N_j}{\partial \xi}\right)+\left(a\frac{\partial N}{\partial \xi}-b\frac{\partial N_i}{\partial \eta}\right)D_5\left(d\frac{\partial N_j}{\partial \eta}-b\frac{\partial N_j}{\partial \xi}\right)\right)$$

$$K_{22}=\frac{1}{|J|^2}\left(\left(d\frac{\partial N_i}{\partial \eta}-c\frac{\partial N_i}{\partial \xi}\right)D_4\left(d\frac{\partial N_j}{\partial \eta}-c\frac{\partial N_j}{\partial \xi}\right)+\left(a\frac{\partial N_i}{\partial \xi}-b\frac{\partial N_i}{\partial \eta}\right)D_5\left(a\frac{\partial N_i}{\partial \xi}-b\frac{\partial N_i}{\partial \eta}\right)\right)$$

3.3 Isoparametric Elements

Problems involving curved boundaries cannot be modelled satisfactorily by using straight-sided elements. A new generation of powerful elements, known as 'isoparamatric elements' has been developed for this purpose. The isoparamatric elements were introduced by Irons [1966] and was well documented by Zienkiewicz [1979].

The isoparamatric concept allows any arbitrary geometry to be closely approximated, thereby minimising any error associated with modelling the geometry and without resorting to the use of the fine mesh. This is because the geometry of the edges of an element will vary in the same way as the displacement of shape function. To derive the isoparamatric element equations, we first introduce a local or natural coordinate system for each element shape. Then, the shape functions will have to be expressed in terms of the local coordinates. The representation of geometry in terms of shape functions can be considered as a mapping procedure which transforms a regular shape, like a straight-sided triangle or rectangle in local coordinates system, into a distorted shape like a curved-sided triangle or rectangle in the global cartesian coordinate system. Lagrange polynomials are often used for the construction of shape function of elements. The basic form of the Lagrange polynomial in a single coordinate system with n nodes is

$$f(x) = \sum_{i=1}^{n} \ell_i^{\,n}(x)\, f_i$$

where $\ell_i^n(x)$ is called the Lagrange multiplier function and is given by

$$\ell_i^n(x) = \frac{(x-x_0)(x-x_1)----(x-x_n)}{(x_i-x_0)(x_i-x_1)--(x_i-x_n)} \qquad (3.3.1)$$

It is obvious that $\ell_i^n(x)$ possesses that properties of

$$\ell_i^n(x_k) = \begin{matrix} 0 & k \neq i \\ 1 & k = i \end{matrix}$$

and thus fits in with the definition of shape function.

To formulate rectangular 'isoparamatric' element as shown in figure (3.2.3), it is usually most convenient to make the function dependent on nodal values placed on the element boundary, which is called the 'Serendipty family'. Therefore, to construct a shape function for the isoparamatric element (see figure (3.3.7)), it is necessary to shift the coordinate origins from the left end to the centre and to change the variable from x/ℓ to $(1+\xi)/2$. So, for the linear element, which has 4-corner nodes, apply the Lagrange polynomial (3.3.1) as $N = \ell_i^n(\xi)\ell_j^{\,n}(\eta)$ with reference to figure (3.3.7), then

(1) at node i, $\xi_i = -1$ $\eta_i = -1$

the shape function

$$N_i = \frac{1}{4}(1-\xi)(1-\eta) \qquad (3.3.2)$$

(2) at node j, $\xi_j = 1$, $\eta_j = -1$

$$N_j = \frac{1}{4}(1+\xi)(1-\eta) \qquad (3.3.3)$$

(3) at node k, $\xi_k = 1, \quad \eta_\xi = 1$

$$N_k = \frac{1}{4}(1+\xi)(1+\eta) \qquad (3.3.4)$$

(4) at node ℓ, $\xi_\ell = -1, \quad \eta_\ell = -1$

$$N_\ell = \frac{1}{4}(1-\xi)(1+\eta) \qquad (3.3.5)$$

or using i as the general subscript for all nodes,

$$N_i = \frac{1}{4}(1+\xi\ \xi_i)\ (1+\eta\ \eta_i)$$

in which ξ_i and η_i are the coordinate values of node i. By letting $\xi_o = \xi\ \xi_i$ and $\eta_o = \eta\ \eta_i$

then we can have

$$N_i = \frac{1}{4}(1+\xi_o)(1+\eta_o)$$

With these values of equations (3.3.2) to (3.3.5), we can find the general coordinates in terms of (x,y) i.e.

$$x_j = \sum_{i=1}^{4} N_i\ (\xi,\eta)\ x_i$$

and

$$y_i = \sum_{i=1}^{4} N_i\ (\xi,\eta)\ y_i \qquad (3.3.6)$$

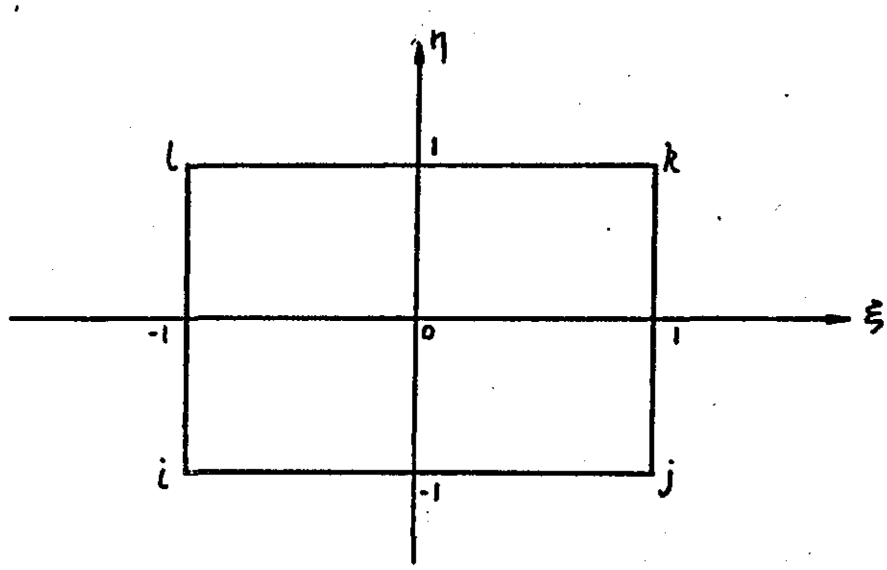


Figure (3.3.7)

Similarly, we can formulate the shape function for the quadratic 'rectangular' elements with 8-nodes, where it has three nodes along one edge.

(i) Corner nodes

$$N_i = \frac{1}{4}(1+\xi_o)(1+\eta_o)(\xi_o+\eta_o)$$

(ii) Mid-side nodes

$$\xi_i = 0 \qquad N_i = \frac{1}{2}(1-\xi^2)(1+\eta_o)$$

$$\eta_i = 0 \qquad N_i = \frac{1}{2}(1+\xi_o)(1-\eta^2)$$

We can now formulate triangular 'isoparamatric' elements by constructing appropriate shape functions. For the three-noded triangular element as shown in figure (3.3.8)

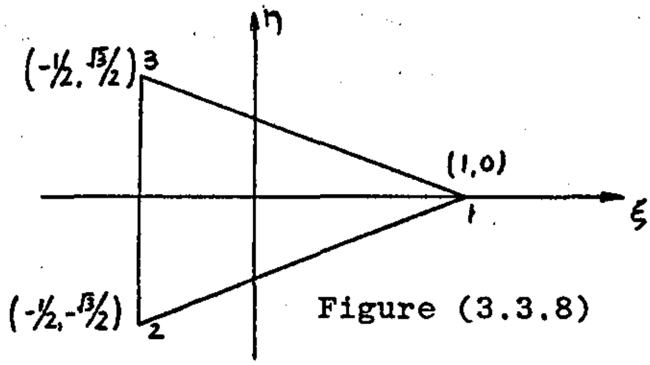


Figure (3.3.8)

the shape functions in terms of the local coordinates are

$$N_1(\xi,\eta) = \frac{1}{3}(1+2\xi)$$

$$N_2(\xi,\eta) = \frac{1}{3}(1-\xi-\sqrt{3}\eta)$$

$$N_3(\xi,\eta) = \frac{1}{3}(1-\xi+\sqrt{3}\eta) \qquad (3.3.9)$$

With these values of equations (3.3.9), we can find the general coordinates in terms of (x,y), i.e.

$$x = \sum_{i=1}^{3} N_i(\xi,\eta)\, x_i$$

$$y = \sum_{i=1}^{3} N_i(\xi,\eta)\, y_i$$

Similarly, for the quadratic 'triangular' elements with 6-nodes, where it has 3-nodes along one edge. The shape functions in terms of the local coordinates are:-

(i) Corner nodes

$$N_1 = (2L_1 - 1)\, L_1 \qquad \text{etc}$$

(ii) Mid-side nodes

$$N_4 = 4\, L_3\, L_5 \qquad \text{etc}$$

where L_1, L_3 and L_5 are defined by the area coordinates such as

$$L_1\ (\xi, \eta) = \frac{1}{3}\ (1+2\xi)$$

$$L_3\ (\xi, \eta) = \frac{1}{3}\ (1-\xi-\sqrt{3}\eta)$$

$$L_5\ (\xi, \eta) = \frac{1}{3}\ (1-\xi+\sqrt{3}\eta)$$

3.4 Plate-bending Elements

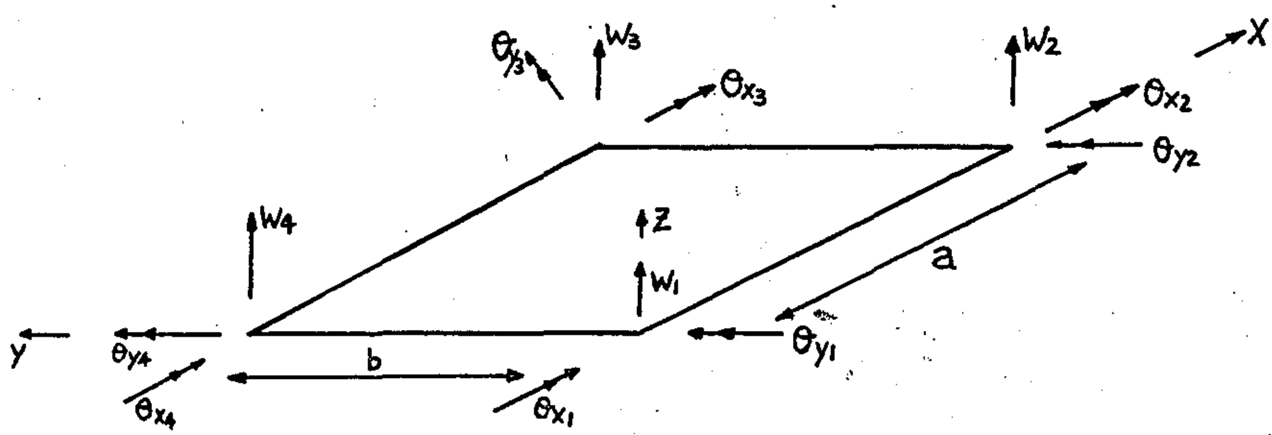


Figure (3.4.1)

The plate-bending element has three degrees of freedom at each node θ_x, θ_y and w respectively as shown in figure (3.4.1). The thickness of the plate is assumed to be small compared to its other dimensions and the deflection of the plate under load is assumed to be small compared to its thickness. The state of displacement at any point within the element may be represented by three components.

$$\left\{\delta(x,y)\right\} = \begin{Bmatrix} w_i \\ \theta_{yi} \\ \theta_{xi} \end{Bmatrix}$$

The assumed displacement field for the plate element is

$$\begin{Bmatrix} u \\ \nu \\ w \end{Bmatrix} = \sum_{i=1}^{m} \begin{bmatrix} 0 & ZN_i & 0 \\ 0 & 0 & -ZN_i \\ N_i & 0 & 0 \end{bmatrix} \begin{Bmatrix} w_i \\ \theta_{yi} \\ \theta_{xi} \end{Bmatrix} \qquad (3.4.2)$$

where the shape functions N_i are small as plane-stress elements and coordinate Z originates at the mid-surface. There is no need for isoparamatric coordinate $\mathcal{L}$, because Z and $\mathcal{L}$ have identical directions. The strain-displacement matrix [B] is similar to previous section (3.2).

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \begin{bmatrix} \partial u/\partial x \\ \partial \nu/\partial y \\ \partial u/\partial y+\partial \nu/\partial x \\ \partial \nu/\partial z+\partial w/\partial y \\ \partial u/\partial z+\partial w/\partial x \end{bmatrix} \qquad (3.4.3)$$

$$\begin{Bmatrix} \partial u/\partial\xi \\ \partial u/\partial\eta \\ \partial u/\partial z \\ \partial v/\partial\xi \\ \partial v/\partial\eta \\ \partial v/\partial z \\ \partial w/\partial\xi \\ \partial w/\partial\eta \\ \partial w/\partial z \end{Bmatrix} = \sum_{i=1}^{m} \begin{bmatrix} 0 & Z\,\dfrac{\partial N_i}{\partial\xi} & 0 \\ 0 & Z\,\dfrac{\partial N_i}{\partial\eta} & 0 \\ 0 & N_i & 0 \\ 0 & 0 & -Z\partial N_i/\partial\xi \\ 0 & 0 & -Z\partial N_i/\partial\eta \\ 0 & 0 & -N_i \\ \partial N_i/\partial\xi & 0 & 0 \\ \partial N_i/\partial\eta & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (3.4.4)$$

Therefore

$$\left\{\frac{\partial u}{\partial x},\ \frac{\partial u}{\partial y}\ \ldots\ldots,\ \frac{\partial w}{\partial z}\right\} = \begin{bmatrix} [J]^{-1} & 0 & 0 \\ 0 & [J]^{-1} & 0 \\ 0 & 0 & [J]^{-1} \end{bmatrix} \left\{\frac{\partial u}{\partial\xi},\ \frac{\partial u}{\partial\eta},\ldots\frac{\partial w}{\partial z}\right\} \qquad (3.4.5)$$

Where $[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & 0 \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & 0 \\ 0 & 0 & 1 \end{bmatrix}$

and det $[J] = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta} = \sum_{i=1}^{m} \frac{\partial N_i}{\partial \xi} x_i \sum_{i=1}^{m} \frac{\partial N_i}{\partial \eta} y_i -$

$$\sum_{i=1}^{m} \frac{\partial N_i}{\partial \xi} y_i \sum_{i=1}^{m} \frac{\partial N_i}{\partial \eta} x_i$$

The inverse $[J]^{-1}$ matrix is

$$[J]^{-1} = \frac{1}{\det[J]} \begin{bmatrix} \frac{\partial y}{\partial \eta} & -\frac{\partial y}{\partial \xi} & 0 \\ -\frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \xi} & 0 \\ 0 & 0 & \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\right) \end{bmatrix} \qquad (3.4.6)$$

To find the matrix [B], we have to use $[J]^{-1}$

With equation (3.4.3), (3.4.4) and (3.4.5). We can yield

$$\{\underline{\xi}\} = [B]\ \{\underline{\sigma}\}$$

i.e. $$\begin{Bmatrix} \xi_x \\ \xi_y \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix} = \sum_{i=1}^{m} \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \beta_i & 0 & -N_i \\ \alpha_i & N_i & 0 \end{bmatrix} + \begin{bmatrix} 0 & Z\alpha_i & 0 \\ 0 & 0 & -Z\beta_i \\ 0 & Z\beta_i & -Z\alpha_i \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \right) \begin{Bmatrix} w_i \\ \theta_{yi} \\ \theta_{xi} \end{Bmatrix}$$

(3.4.7)

Let,

$$\alpha_i = J^{-1}_{11} \frac{\partial N_i}{\partial \xi} + J_{12}^{-1} \frac{\partial N_i}{\partial \eta}$$

$$= \frac{1}{|J|}\left[\sum_{i=1}^{m}\left(\sum_{i=1}^{m} y_i \frac{\partial N_i}{\partial \eta} \frac{\partial N_i}{\partial \xi} - \sum_{i=1}^{m} \frac{\partial N_i}{\partial \xi} y_i \frac{\partial N_i}{\partial \eta}\right)\right] = \frac{1}{|J|}\sum_{i=1}^{m}\left(a \frac{\partial N_i}{\partial \xi} - b \frac{\partial N_i}{\partial \eta}\right)$$

$$\beta_i = J_{21}^{-1} \frac{\partial N_i}{\partial \xi} + J_{22}^{-1} \frac{\partial N_i}{\partial \eta}$$

$$= \frac{1}{|J|}\left[\sum_{i=1}^{m}\left(\sum_{i=1}^{m} x_i \frac{\partial N_i}{\partial \xi} \frac{\partial N_i}{\partial \eta} - \sum_{i=1}^{m} x_i \frac{\partial N_i}{\partial \eta} \frac{\partial N_i}{\partial \xi}\right)\right] = \frac{1}{|J|}\sum_{i=1}^{m}\left(d \frac{\partial N_i}{\partial \eta} - c \frac{\partial N}{\partial \xi}\right)$$

The entire matrix [B] may be split into a part $[B_0]$ independent of Z and a part Z $[B_1]$ linear in Z as in equation (3.4.7). The stiffness matrix is therefore given by

$$\left[K_{rs}^{b}\right] = \int_{vol} [B]^T [D] [B] \; dx\, dy\, dz$$

$$= \int_{vol} [B_0 + ZB_1]^T [D] [B_0 + ZB_1] \; dx\, dy\, dz$$

Integration with respect to z is between the limits $\pm$ t/2, because of the distribution of zeros in $[B_0]$, $[B_1]$ and [D], products $Z[B_0]^T[D][B_1]$ and $Z[B_1]^T[D][B_0]$ vanish, $[B_0]^T[D][B_0]$ involves only the 'transverse shear' portion of [D]. Therefore,

$$\left[K_{rs}^{b}\right] = \int_{area} [B]^T [D] [B] dx\, dy = \int_{area} [B]^T [D] [B] \det [J] \; d\xi\, d\eta$$

Where the matrix [D] is

$$[D] = \begin{bmatrix} D_1 & D_2 & 0 & 0 & 0 \\ D_3 & D_4 & 0 & 0 & 0 \\ 0 & 0 & D_5 & 0 & 0 \\ 0 & 0 & 0 & D_6 & 0 \\ 0 & 0 & 0 & 0 & D_7 \end{bmatrix}$$

Where $D_1 = D_4 = \dfrac{Et^3}{12(1-\nu^2)}$, $D_2 = D_3 = \dfrac{E\nu t^3}{(1-\nu^2)}$

$D_5 = \dfrac{Et^3}{2(1+\nu)}$, $D_6 = D_7 = \dfrac{(1-\nu)Et^3}{2K(1-\nu^2)}$

So we have

$$\left[K^b\right] = \int\int \quad B^T \; D \; B \; \det \; [J] \; d\xi \; d\eta$$

Where

$$B^TDB = \begin{bmatrix} \beta_i \; D_6 \; \beta_i + \alpha_i D_7 \alpha_i & \alpha_i \; D_7 \; N_i & -\beta_i \; D_6 \; N_i \\ & \alpha_i D_1 \alpha_i \; + \; \beta_i D_5 \beta_i + N_i D_7 N_i & -\beta_i D_5 \alpha_i - N_i D_7 N_i \\ \text{Symmetry} & & \\ & & -\beta_i D_4 \beta_i + \alpha_i D_5 \alpha_i - N_i D_6 N_i \end{bmatrix}$$

3.5 Shell-type Elements

The first application of the finite element method to shell structures was made by representing the curved surface of the shell by flat elements, either triangular or quadrilateral in shape, and superimposing the plane-stress and bending stiffness matrices disregarding the coupling effect within each element. This approach was first presented by Zienkiewicz and Cheung [1965] and Clough and Tocher [1965] by using displacement type flat element.

$$[K] = \left[\begin{array}{cc|ccc|c} \multicolumn{2}{c|}{\multirow{2}{*}{K^p_{rs}}} & 0 & 0 & 0 & 0 \\ & & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & & & & 0 \\ 0 & 0 & & K^b_{rs} & & 0 \\ 0 & 0 & & & & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right] \qquad (3.5.1)$$

where $\left[K^p_{rs}\right]$ is a submatrix of in-plane action and $\left[K^b_{r,s}\right]$ is a submatrix of bending action.

The stiffness matrix shown in (3.5.1) used a system of local coordinates as the in-plane, and bending components. The formulation of the shell-type elements is well documented in Zienkiewicz [1979] The shell-type elements have been used commonly at Austin Rover for constructing car bodies.

3.6 Datum Transformation and Assemblage of Element Equations

The various methods of deriving element characteristic matrices and vectors have been described in the above sections. Before proceeding to consider how these element matrices and vectors are assembled to obtain the characteristics of the entire system of elements, we need to derive a technique to enable us to transform the local coordinates to a common global system.

The element characteristics are computed in local coordinate systems suitably oriented for minimising the computational effort. The local coordinate system may be different for different elements. When a local coordinate system is used, the directions of the nodal degrees of freedom will also be taken in a convenient manner. Therefore, before the element equations can be assembled, it is necessary to transform the element equations derived into local coordinate so that all the element equations are referred to a common global coordinate system.

If $\underline{\delta}_L^i$ and $\underline{\delta}^j$ represent the vectors of local and global displacements, then the transformation between the two may be written as

$$\underline{\delta}_L^i = [L]\ \underline{\delta}^j$$

where $[L]$ is a 6x6 matrix comprised of two 3x3 matrices denoted by $[\lambda]$

as $$[L] = \begin{bmatrix} \lambda & o \\ o & \lambda \end{bmatrix}$$

The matrix $[\lambda]$ is formed of direction cosines of the angles formed between the two sets of axes:-

$$[\lambda] = \begin{bmatrix} \lambda_{x_L x} & \lambda_{x_L y} & \lambda_{x_L z} \\ \lambda_{y_L x} & \lambda_{y_L y} & \lambda_{y_L z} \\ \lambda_{z_L x} & \lambda_{z_L y} & \lambda_{z_L z} \end{bmatrix}$$

where λ_{xLx} is the cosine of the angle between the x_L and x-axis, etc.. The same transformation holds for nodal forces, and for a complete element may write

$$\underline{\delta}_L^e = [T]\ \underline{\delta}^e \qquad (3.6.1)$$

and

$$\underline{F}_L^e = [T]\ \underline{F}^e \qquad (3.6.2)$$

The local coordinate system of the element equation can be expressed in the standard form

$$\left[K^e\right]\ \underline{\delta}^e = \left[F^e\right] \qquad (3.6.3)$$

by substituting (3.6.1) and (3.6.2) into (3.6.3), we obtain

$$\left[K^e\right]\ \left[T^e\right]\ \underline{\delta}^e = \left[T^e\right]\ \underline{\delta}^e \qquad (3.6.4)$$

premultiplying this equation throughout by $\left[T^e\right]^{-1}$ yields

$$\left[K^e\right] = \left[T^e\right]^T \left[K_L^e\right] \left[T^e\right]$$

Once the element characteristics, namely the element matrices and element vectors are found in a common global coordinate system, the next step is to assemble all the element matrices to obtain a master stiffness matrix and a total applied vector. The procedure of assembling the element matrices and vectors is based on the requirement of 'compatibility' at the element nodes. This means that at the nodes where elements are connected, the values of the unknown nodal degrees of freedom are the same for all the elements joining at the node. The assembling procedure is the same regardless of the type of problem and the number and the type of elements used.

In structural problems, the nodal variables are usually generalised displacements which can be translations, rotations, curvature or other spatial derivatives of translations. When the generalised displacements are matched at a common node, the nodal stiffness and nodal loads of each of the elements sharing the node are added to obtain the net stiffness and the net load at that node.

In many practical problems, the element matrix $[k^e]$ and the master matrix $[K]$ are symmetrical and we therefore only have to assemble the lower (or the upper) triangle of $[K]$ using the lower (or upper) triangle of $[k]$.

From the result in (2.4) of the cantilever beam, we can easily calculate the analytical solutions and make comparison with computational results to confirm the accuracy. However, if the structure is more complicated, i.e. a car body, then the problem becomes more difficult to solve.

In the case of symmetrical structure, it is possibly the analysis by modelling the one half of the structure and applied appropriate constraints at the plane of symmetry. For example, take a symmetrical beam element, where the non-symmetrical loadings are applied as shown in (3.6.5)

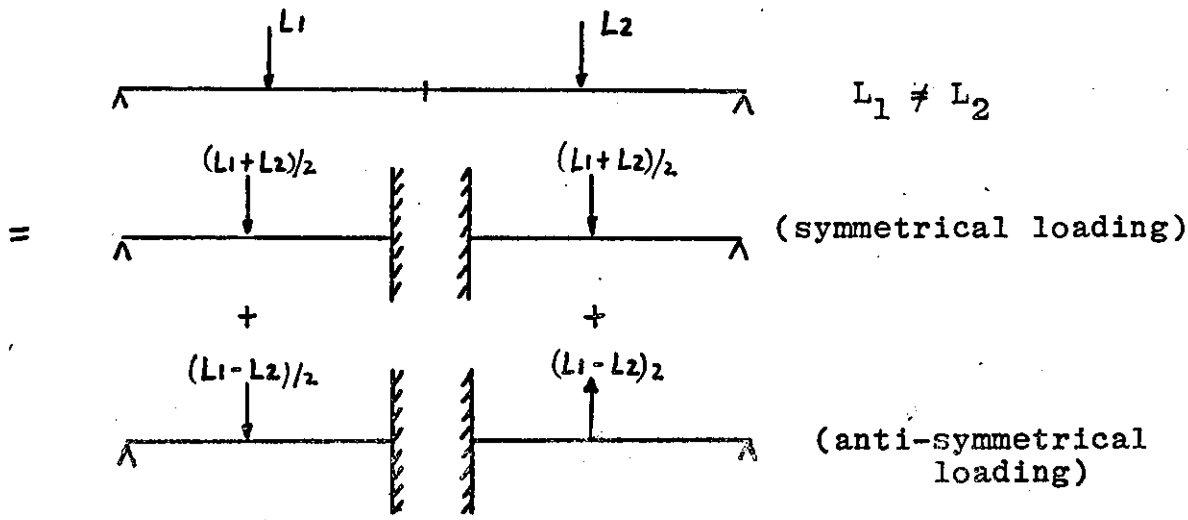


Figure (3 .6.5)

In this case, we have to apply two different loadings at half of the structure separately with the appropriate constraints. The two different loadings are symmetrical and anti-symmetrical. The results of these two loadings are added together to give the equivalent result of the half-sized structure.

The car body, as we know, is non-symmetrical, but the capacity of the computer storage is not adequate to cope with the full-sized model. Generally, we are restricted to carry out the finite element on one-half of the car body.

The substructure or superelement method is the technique that enables the problem to be reduced to a more manageable size. Basically, the method is based on subdividing the large structure into smaller parts which are analyzed separately to obtain relationships between forces and displacements. Technically, the superelement method is expressed by the generalized coordinates associated with the nodes in the interior of each substructure on its boundaries, so that reduced matrices associated only with the generalized coordinates of the boundary nodes can be developed. This procedure is called static condensation. The advantage of the superelement method is that it enables the division of the entire structure under consideration into several parts which are connected to each other by a finite number of nodes and the interelement boundaries between these nodes. Each part may itself have many elements. The element matrices of each part are assembled into a submaster matrix for the part. These submaster matrices for each part are just like the element matrices for an element; each part can thus be viewed as a superelement, (for more details, see chapter 4). Several finite element programs such as NASTRAN [1976], ASKA [1971], SESAM-69 [1974], GIFTS [1973] all adopt the above method.

4 INTERFACE PROBLEMS

4.1 Introduction

Many automatic mesh generation techniques have been developed to generate finite element models, (see chapter 5 for details), but none appear suitable for a complicated structure like a car body. The manual preparation of an input data file for the finite element analysis of a structure can be extremely time-consuming in all but simple examples. It has only been possible to construct a crude model in such a way, because the user could not handle the amount of data required. An interactive data-handling system using a digitizer via a graphics facility has proved capable of handling large amounts of data and saved labour and cost, but it still only provides a crude geometrical description of the structure.

The purpose of this chapter is to give the general concepts of this project, and an overview is given in Figure (4.2). At present, the Austin Rover group (ARG) is able to hold a detailed geometric description of a car body and its internal structures in a Gemometric Data Base (GDB). The aim of this project is to develop an automatic mesh generation program which will use the geometric data, together with supplementary structure data, to generate a finite element model. As an intermediate stage, therefore, it is proposed to create a new file, namely the Structural Data Base (SDB).

It is difficult to mesh a vehicle in detail without breaking down the model into a number of components. Also, it is inefficient to re-run an entire model for analysis when changes in only a small portion are to be investigated. Therefore, substructuring the vehicle would be the way to identify and describe various portions of the model.

Before the automatic mesh generation program can be used, we have to supply an algorithm to ensure all the data in SDB are correctly defined. The algorithm consists of checking that the substructures are geometrically compatible and fit together with no gaps between them. We must also ensure that the boundaries of the substructures are continuous after introducing thickness into the structure (see section 4.4).

NASTRAN (NAsa STRuctural ANalysis) is the finite element program used by the Austin Rover Group to analyse the finite element model. The program allows the construction of 'superelements' which store all the substructures by reducing all the internal nodes data to boundary nodes data. In this way, we can overcome the problem of computer storage and allow the model to be meshed with great accuracy.

The ARG uses only isoparametric elements and we therefore restrict ourselves to the use of these elements in the automatic mesh generation program. In addition, shell-type elements are preferred, since it is found to be a more predictable tool in the analysis of a car structure.

Finally, we have to implement the automatic mesh generation program to mesh individual panels and assemble them into relevant substructures. All data will be stored in a universal format since this is the way to input all the data to NASTRAN for analysis (see section 4.6.2). Since the car model will be accurately represented by the superelement technique, accurate analysis will be possible.

After the mesh generation program has completely meshed the structure, we are required to build in an editing routine that allows the user the options of changing data in the SDB.

4.2 Overview Flowchart

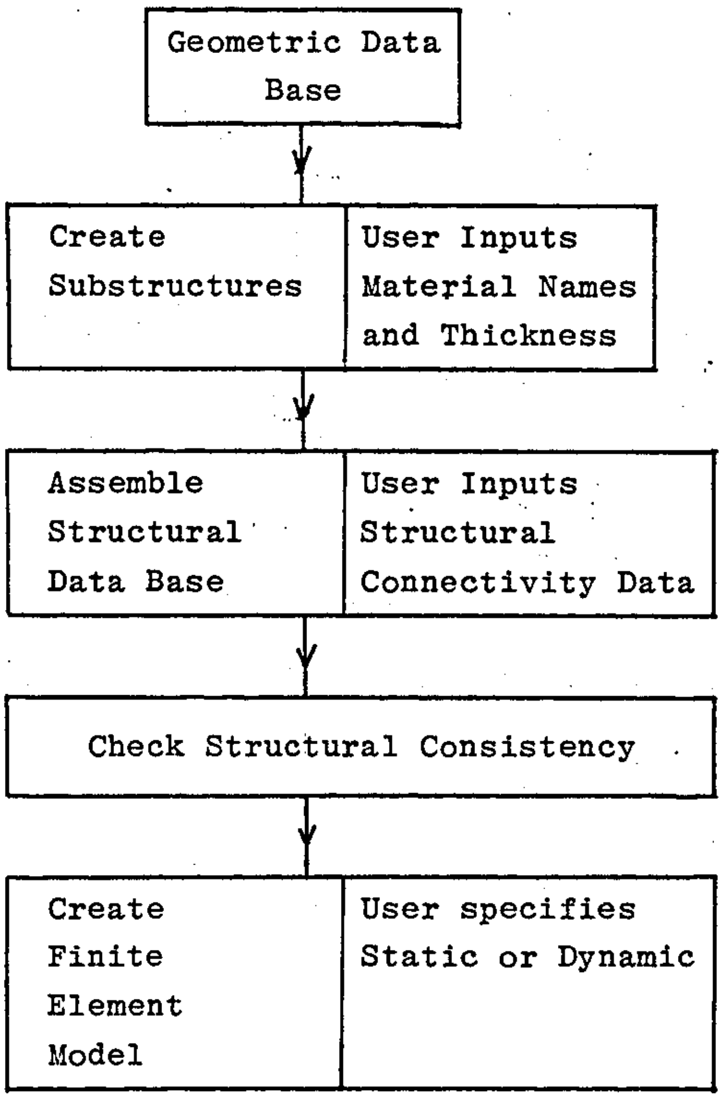


Figure (4.2)

4.3 Geometry Definition

4.3.1 Geometric data base

In the past, there have been difficulties in obtaining accurate geometrical descriptions of the car model, because the computer facility and the mathematical techniques have not been capable of giving accurate descriptions of the car body. In this case, manual involvement has played a large part in the process of obtaining geometrical description of the model. Therefore, the initial stage is to have an automatic generation of input data of the car structure by the computer with minimal human involvement. Presently, the Austin Rover Group are able to hold within the computer a geometrical description of the shape of the car and its internal structure. The data is subsequently used to aid various design and manufacturing processes. The company uses 'CONSURF' very successfully for the geometrical description of the external body shape. Now, they are applying similar techniques with the aid of interactive graphic facilities such as CATIA and CADAM to construct the geometrical description of the internal structures. The whole structure is broken down into many panels and mathematically described by polynomials stored in the Geometrical Data Base (GDB).

4.3.2 Structuring the car body

It is difficult to construct in detail a finite element model of the car body, without breaking down the model into a number of components and it is inefficient to re-run an entire model for analysis when changes in only a small portion are to be investigated. A car model is substructured to provide a natural way to identify and describe various portions of the vehicle. These substructures are used in the fabrication and assembly processes, and the conceptual and mathematical models of each substructure can then be checked independently of the others. This means than an engineer can analyse his own particular substructure, while also contributing to the overall model.

All the geometrical data is described in terms of panels, but the substructures may be defined by more than one panel. Therefore, we require the user to input the names of each substructure and collect all the relevant panels to it.

4.4 Assembly of the structural data base

The general idea of this project is to make use of the geometrical data for the construction of a finite element model.. Also, if the physical properties of the finite element model, such as the thickness and material names are known, we can analyse the behaviour of the model. Therefore, we require the user to specify these physical properties and also the connectivity of the different substructures. In addition we may require the user to input all the discrete loading positions, since the positions can be defined in advance.

The Structural Data Base (SDB) is used to store all the geometrical data and the additional information, because the geometrical data in the GDB is not to be manipulated and changed.

4.5 Checking of structural consistency

Before the automatic mesh generation program can be used, we have to introduce an algorithm to ensure that all the information in the SDB is correctly defined. The algorithm consists of checking that the connectivities of substructures are geometrically compatible, by ensuring there are no gaps between substructures and that they are connected correctly. One difficulty arises when it is used to check for consistency between boundaries of panels within a substructure during assembly. The reason is that the geometrical data is an assembly of surfaces which have no physical properties. When we include these properties, such as the thickness, into the finite element model, it can result in the cross-sections of the substructures taking a variety of irregular forms. These include open multiple flange and single and double-celled closed cross-sections. These sections are generally made of folded parts of uniform thickness, joined at tails. However, each part may have a different thickness, so the algorithm is required to ensure that varying thicknesses of different parts do not affect the smoothness of the boundaries when they are assembled together.

4.6 Creation of Finite Element Model

4.6.1 Preliminary considerations

The objective of this project is to develop a mesh generation system which automatically generates a finite element model based on the structure's geometric properties. We know the internal structure contains many complicated shapes. Therefore, it will make computation simpler by breaking down each substructure into many regions for meshing. The technique will be described in section (4.6.2).

Although the process is automatic, the user should have the option to overrule at various input stages. At this stage, we require the user to specify the type and the number of elements on each panel. The reason is that there are difficulties in determining the optimum number of elements on each panel and to provide the analytical solution of the model. Therefore, we are aiming to mesh each panel with the minimum number of elements. To achieve thds objective, we may require the user to input this information to the SDB. In order to do this, we are assuming the user has the experience of using the finite element technique on similar structures. We can take advantage of their experience to decide on the type and the number of elements on each panel.

There will be difficulties in deciding how many elements are required to model the stiffness of a particular structure. In general, the greater the density of elements in a model, the better the numerical approximation to the true solution. The actual number of finite elements necessary to model a given structure is a compromise depending on the nature of the structure, its loading and supports, the type of elements used, and the purpose for which the results of the analysis are to be used. We can use the number of elements based on the previous models as a guideline which can then be built into the program to give a warning to the user. Most of the internal structures are governed by the geometry and the boundary conditions. In addition, these restrictions can also be used to decide on the number of elements on each panel.

There will be difficulties in ensuring that the number of elements specified by the user are meshed exactly onto the panels. Unlike the external panels, which are generally smooth, the inner panels contain numerical cutouts, bosses and ribs. Additional modelling complications are encountered in the cases where the inner and outer panels are adhesively bonded to form a sandwich-like structure. Therefore, complications will arise in dividing the surfaces into 'smooth' areas for meshing. If the panels are meshed with more elements than required, we may introduce computational errors which will result in a loss of accuracy.

4.6.2 <u>'NASTRAN' modelling techniques</u>

NASTRAN is the general purpose finite element program used at the Austin Rover Group for analysis of the finite element model. The program is based on the displacement method of finite element analysis and it has the facility to collect many of the elements and reconstruct them into a 'superelement' by condensing all the internal nodes into boundary nodes. The condensation technique is introduced by Guyan [1965], and is called the Guyan Reduction in the NASTRAN computer program. The technique enables us to mesh each panel in detail and represent it by a superelement. These panels are then assembled into sub-structures. Each substructure can also be formed into a superelement such as superelement-within-superelement. In this way, we can have the entire floor of the vehicle represented as one superelement whose mass and stiffness properties can be defined in terms of displacements at the periphery of the floor. This description would be combined with similar superelement models of the roof, wheel wells, fenders etc., in the same way that finite elements are combined to model the floor itself. By assembling the model according to increasing levels of complexity, we are still able to change portions of the structure without effecting the connecting substructure models. Also this technique is attractive since it parallels the design process where major structural components, or substructures, are often designed by different engineering groups or at different times. It is desirable, therefore, to use a substructure approach so that such designs and modifications may proceed as independently as possible, with due considerations being given to the coupling of substructures to form the complete structure.

In the past, the finite element model of the car has been constructed in halves, rather than in full-sized models. The car body could be represented symmetrically, because the components which make a car structure non-symmetric, such as the steering-gear, air-conditioning equipments, transversely mounted engine, spare-type, and other cargo, could be easily identified. The influence of these non-symmetric components may be economically studied by making mass and stiffness modifications to the base system represented by its symmetric and anti-symmetric modes. The reason is that the storage capacities of the computer can be exceeded if the full size finite element structure of the car body is used for analysis, as the total number of degrees of freedom or the size of the stiffness matrices required becomes larger. Now, we can construct a full size finite element model, since the superelement provided by NASTRAN could reduce the size of the model to be stored in the computer memory.

4.6.3 To mesh the structure using isoparametric elements

The Austin Rover Group have been using the isoparametric family of elements to mesh structures. The reason is that these elements have shown great accuracy in approximating any arbitrary geometry closely, thereby minimizing any errors associated with modelling the geometry without resorting to the use of a fine mesh along the boundaries. This has proved particularly useful for meshing internal structures. The program must be based on the isoparametric elements and it must closely fit the surfaces described by the polynomials from the GDB. There will, however be problems in fitting the isoparametric elements to the surfaces, due to the polynomials of the isoparametric elements being usually up to third order. Whereas the polynomials used to describe the surfaces can be higher than third order. Also, we need to consider the minimum size of elements to be meshed, such as the rectangular elements, which are unable to cope with long thin elements. The best results are obtained if the aspect ratio is less than 5:1.

In the past, areas which are complicated have been simplified or modelled by beam elements, because it is very difficult and time-consuming to mesh the complicated area in detail. Also, if the internal structure is meshed by fine grids, it will cause the computer storage to overflow. Thus, due to their geometrical simplicity, beam elements are the most popular to be used in complication areas (e.g. pillars, side-crossmembers etc). However, beam elements have drawbacks for use in design. First they are not a fully predictable tool, since we are required to calculate the second-moment of areas. Secondly, beam models cannot account for sectional distortions under loads, since it is assumed that their cross-sectional shape is fixed. In addition, it is difficult to model some members of a frame, such as spring seats and attachment brackets, which are not beamlike, or to model large holes. Also, it is not convenient to incorporate design changes (e.g. gauge modifications, welding, bolt pattern changes and shape changes) into beam finite element models. Lastly, beam models are not very pictorial, and even graphic displays do not permit detection of any input data errors.

4.7 Computational routine

When the automatic mesh generation program is used to mesh the panels we have to take into account all the boundary conditions, such as the joint positions and the number of nodes used for assembly, to be represented adequately at the boundaries.

Finally, we have to provide an algorithm to ensure that all the panels are filled in with the number of required elements. To do this, we have to check that no nodes or elements are lying outside the boundaries. Thus, we have to make sure that all the spaces are meshed.

All these data will be stored in a standard format, called 'universal', which is commonly used by engineers in this field and it is recognised by the Structural Dynamics

Research Corporation. Also, this is the way that we can input the generated finite element model to NASTRAN for finite element analysis. Actually, we output all the data into the finite element package called 'SUPERTAB', which can be then translated into the NASTRAN format. The advantages of using SUPERTAB is that it provides graphic display facility and checking routines, such as distortion check, minimising bandwidth and renumbering.

Both static and dynamic analysis are carried out in the finite element structures. It is not uncommon to see static analysis being performed on the structures having many thousands of degrees of freedom. It is, however, rare that dynamic analysis be performed on the structures of the same order of complexity, because the cost of such an analysis is so much greater than its counterpart. This actually need not be the case, for often one does not need the same level of discretization for dynamic analysis that is required for static analysis. Thus, it would be desirable to be able to transform a 'large degrees-of-freedom' problem to a more manageable, smaller size one for dynamic analysis, to be performed using the superelement technique. Since a finite number of nodes are used to construct a model, the application of loads and constraints can only be an approximation to the actual distribution. Therefore, the application of an equivalently distributed load is required.

4.8 Editing routine

After we have completely modelled the structure and analysed it, using NASTRAN, the user may be required to change the geometrical data or any geometrical properties.

In this case, we have to implement an editing routine which enables the user to have access to the SDB to make any necessary changes.

5 AUTOMATIC MESH GENERATION TECHNIQUES

5.1 Introduction

Many mesh generation techniques have been derived to the modelling of structures. The reason is that the manual preparation of finite element meshes proves to be cumbersome, costly and prone to errors if the structure is fairly large or geometrically complex. In general the purpose of mesh generation is to produce an accurate representation of the structure with minimum manual involvement.

Applying an automatic mesh generation algorithm to any complicated structure involves many considerations, such as the geometric description of the structure, the type and the number of elements and boundary conditions. It is straight-forward to mesh a simple region. For example, a rectangular region with straight-sided boundaries and mesh with 3x4 elements (say) as shown in Figure (5.1a).

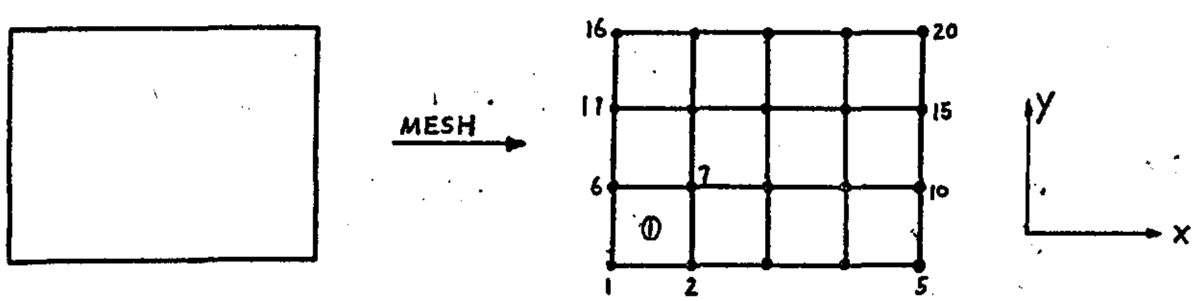


Figure (5.1a) Figure (5.1b)

The simplest type of element used is the linear 'rectangular' element with 4 nodal points at the corners. Since, we are required to mesh with 3x4 elements on the above region. We need to have 5 nodal points on the boundaries in the x-direction and 4 nodal points in the y-direction. The simplest method, and one step above that of specifying the coordinates of every node is the 'straight-line interpolation'. The technique involves specifying the end-point coordinates and node numbers for a straight line; such as $x_6 = 0.0$, $y_6 = 1.0$ for node 6 and $x_{10} = 5.0$, $y_{10} = 1.0$ for node 10. The difference of the nodal point numbers determines the number of divisions on the line. The difference of the coordinates gives the length of the line. The length divided by the number of divisions determines the equal increments of the straight line. These nodal points must then be connected to form elements, and the assemblage of the elements forms the continuous structure. We are required to input the topology of each element. For example, for element 1 it is node 1, node 2, node 6 and node 7 and so on. The program will automatically connect all these topologies to form elements as shown in Figure (5.1b).

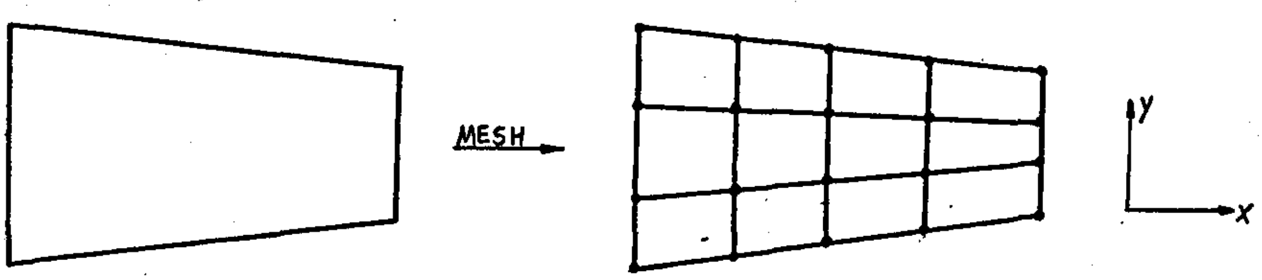


Figure (5.2a) Figure (5.2b)

We can apply the same routine as described above to generate elements on the region as shown in Figure (5.2a). The pattern of elements formed in Figure (5.2b) is similar to Figure (5.1b), except that the area of all the elements are not the same as in Figure (5.2b).

Alternatively, we can apply the isoparametric mapping method, which is described by Zienkiewicz [1967], to mesh the region in Figure (5.2a). The advantage of this method is that it is capable of coping with curved boundaries as shown in Figure (5.3a) as well as generating elements on it as shown in Figure (5.3b).

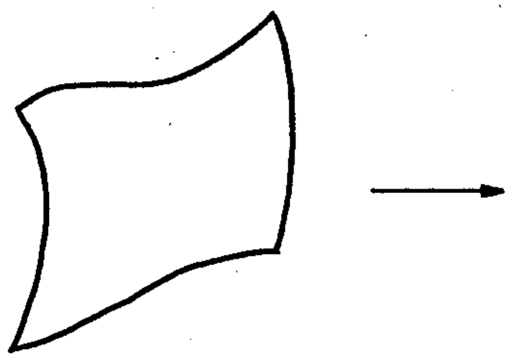


Figure (5.3a)

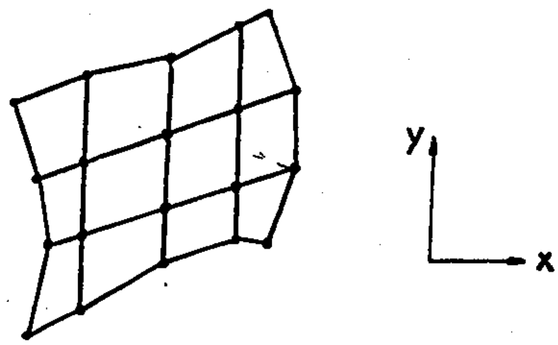
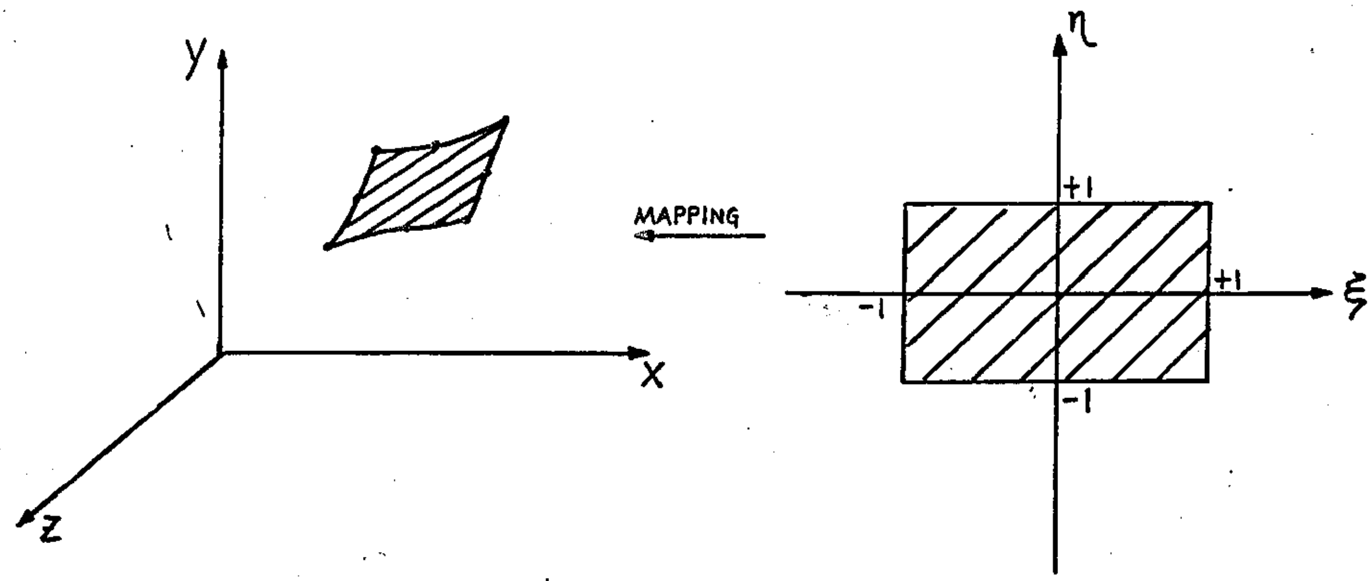


Figure (5.3b)

The isoparametric mapping method is based on polynomial interpolation functions (shape functions) to provide a unique mapping between curvilinear coordinates, (ξ,η) and the cartesian coordinates (x,y) (see chapter 3). The method is not only able to mesh the region in a 2-dimensional plane, but also able mesh region in 3-dimensional space. The method is similar to 2-D by mapping between the cartesian coordinates (x,y,z) and the curvilinear coordinates (ξ,η) as shown in Figure (5.4)



Cartesian Coordinate System Curvilinear Coordinate System

Figure (5.4)

However, the method is unable to directly mesh a region which has an irregular shape as shown in Figure (5.5).

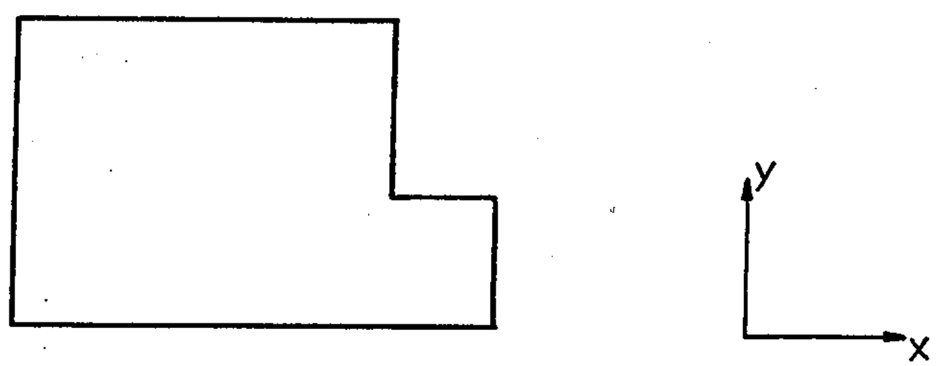
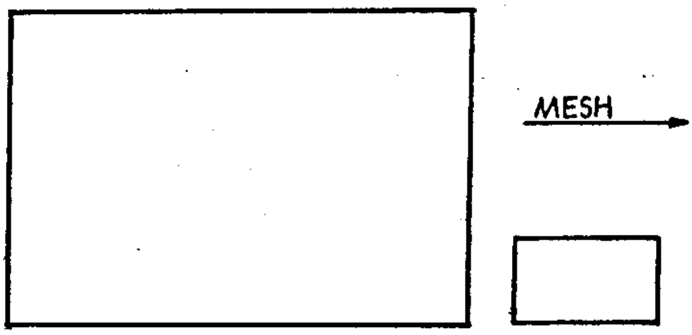


Figure (5.5)

The technique manually subdivides the region into two subregions as shown in Figure (5.6a), then mesh with 3x4 linear elements for the large subregion and 2x2 for the smaller subregion as shown in Figure (5.6b)



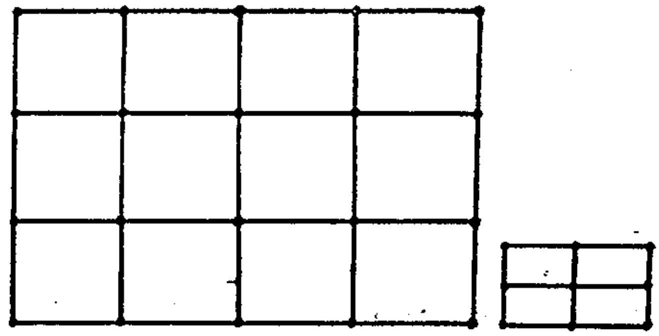


Figure (5.6a) Figure (5.6b)

Alternatively, we can use the random node generation technique to generate nodes on the irregular region, a technique which was originally developed by Suhara & Fukuda [1972]. Finally, we can assemble the two subregions together to restore to the original shape as shown in Figure (5.7).

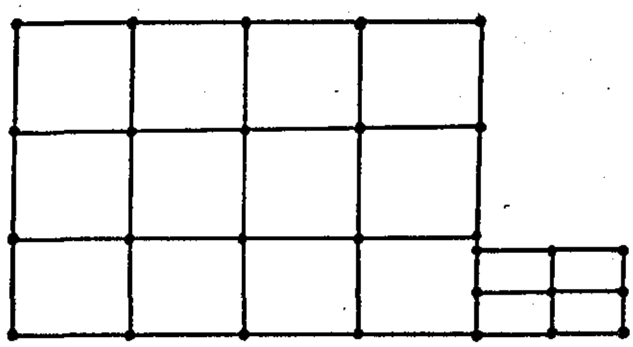


Figure (5.7)

5.2 Considerations for a mesh generation algorithm

The essential feature of the automatic mesh generation program is to mesh the structure accurately. If the structure is a simple region as shown in Figure (5.1); we have no difficulty in generating elements in an regular pattern. However, if the region is an irregular shape as shown in

Figure (5.5), we are required to subdivide the region into subregion so as to enable meshing without causing any computational errors.

Generally, published papers on automatic mesh generation techniques assume that the region is regular. Any irregular shapes are manually subdivided by the user to ensure that the region can be meshed as shown in Figure (5.6).

Ideally, it is best to allow the automatic mesh generation program to choose the type of element and the 'optimum' number of elements on each region to be meshed. However, one problem is the difficulty of knowing the analytical solution over a complex region. This implies that it is difficult to obtain the 'optimum' number of elements on each region and to decide on the type of element to be used. It is possible to find the analytical solution for a simple region such as a square plate. So far, there are no algorithms which have been devised to decide on the type and the 'optimum' number of elements on the complex region to be meshed. We require the user to input the type and the appropriate number of elements and this will depend upon their experience and judgement.

Zienkiewicz & Phillips [1971] have developed an automatic mesh generation scheme for plane and curved surfaces by isoparametric coordinates. The scheme basically requires the user to subdivide the region into quadrilateral zones. Then, the user has to define the number of elements on each zone and the scheme will mesh each zone with 3-nodal triangular element. Durocher & Gasper [1979] have published a a fortran program based on the isoparametric mapping concepts introduced by Zienkiewicz [1971]. The program requires the user to subdivide the region into quadrilateral zones (superelements) with eight sets of nodal coordinates, and to specify the type and the number of elements on each superelement. The program was written based on five commonly used two-dimensional elements, 3-noded triangles, 6-noded straight-sided triangles, 6-noded isoparametric triangles, 4-noded or 8-noded isoparametric quadrilaterals. Fujii & Yuki [1973], Imafuku et al [1980], Ghassemi [1982] and Stefanou [1980] have used a similar technique that required the user to perform a rough division of the considered region into quadrilateral or triangular zones which are further subdivided into triangular or rectangular elements.

5.3 Node generation

Generating nodes on each region is one of the important routines in the mesh generation program. The number of nodes on each region is dependent on the specific number of elements and the type of elements which have been specified at the earlier stage.

The initial stage is to have adequate external nodes to describe the boundaries of a region. The external nodes will be used for defining the grid spacing and the pattern of meshes inside the region. For a simple region with straight-sided boundaries as shown in Figure (5.1a), it is straight-forward to describe the boundaries with four external nodes at the corners. The program will automatically generate the rest of the required external nodes. With these boundary nodes, we have no difficulty in generating interior nodes (see Durocher & Gasper [1979]). Various node and element generation techniques have been reviewed in Buell & Bush [1972] to generate interior nodes and connect these nodes to form elements. The techniques described in the paper assume that the region is smooth and that it is possible to construct elements in a regular pattern. Thacker [1980] has presented an updated survey of the automatic mesh generation techniques with additional information, such as smoothing techniques and recognition of neighbouring points, for modelling irregular region.

There is no difficulty in generating interior nodes on the regular region if all the nodes lie on the straight-lines as shown in Figure (5.1b). However, this can make it difficult to vary element sizes in an economical way, because the number of nodes on each side are restricted to the same number as those on the opposite side. It also creates problem in connecting different region together, because the number of external nodes on common boundary may be different as shown in Figure (5.8).

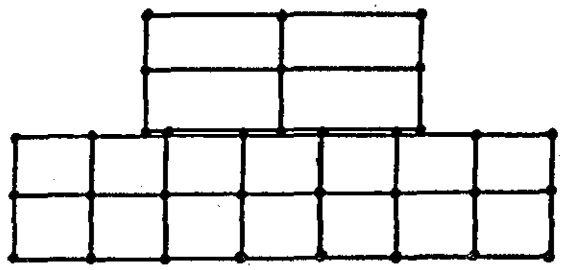


Figure (5.8)

To overcome the problem, we can either manually ensure that the number of external nodes on the connected regions are the same, or insist that the automatic mesh generation program has a routine to connect the region with the different number of nodes on their boundaries (see Ghassemi [1982]).

Sometimes, we are required to analyse a particular part of the region which may need to have a denser meshing than other parts. One way is to allow the user to introduce weighting factors to enable grading of the mesh, (see Zienkiewicz & Phillips [1971] and Durocher & Gasper [1979]).

It is obvious that the shape of the grid will depend on the shape of the boundaries and the portion of the external points. The grid on a region is not always regular, because the external points on the boundaries may not be equal on opposite sides. Also, the grid is dependent on the shape of element used and too much irregularity can effect the accuracy of the computation for which the grid is intended. In this case, for best results, the triangular element must be as equilateral as possible, for acute angles can result in computational instability. Similarly, quadrilateral elements should be as square as possible. Therefore, a checking routine must be provided to ensure that all the elements are not too irregular (see Frederick et al [1974], Lewis & Robinson [1978]).

If the geometry of a region involves curved boundaries as shown in Figure (5.3a), we require a technique to closely fit the external nodes onto the boundaries. Since we are interfacing CAD/FEM, as pointed out in Chapter 1, the external nodes can be automatically generated on the boundaries. The reason is that both functions are polynomials. Alternatively, we can manually input all the required external nodes on the boundaries. Zienkiewicz & Phillips [1971] requires the user to specify the boundary nodes, if curved boundaries are involved, and the number of subdivisions in ξ and η directions. The user can input the weighting factors to grade the mesh. Durocher & Gasper [1979] have developed a program based on some of the suggestions above, such as allowing the program to calculate the interior nodes, if the region is straight-sided. Alternatively, the program allows the user to have the option of inputting the interior nodal coordinates in order to grade the mesh within the given region. One of the limitations of the Durocher's program is that it cannot be efficiently used for triangular region. Wu [1980] has tested the program with some corrections for constructing triangular elements. Ghassemi [1982] has provided a similar method as mentioned above with a Fortran program, except that it is restricted to be used on 3-noded or 6-noded triangular 'isoparametric' elements. The program generates the interior nodes by assuming uniform subdivisions in the plane triangle. The program also contains a 'merge' algorithm to connect different triangular regions together. It allows the user to number different zones in any order.

The isoparametric mapping method is the most popular method used for automatric mesh generation. There are other methods used for generating nodes and connecting nodes to form elements. The transfinite mapping method has been used in papers, such as Jones [1974] and Haber et al [1982] for mesh generation. The method was originally developed by Gordon & Hall [1973] to calculate the nodal coordinates for approximation volumes and surfaces. The method is based on the bivariate blending-function interpolation for producing well-behaved maps from canonical regions such as the unite square to simply connected domain.

Jones [1974] has written a two-dimensional mesh generation program 'Qmesh' based on the scheme developed by Gordon & Hall [1973] to produce quadrilateral elements. A smoothing technique was introduced to produce improved meshes without user interaction. The technique is based on the Laplacian method and is easy to implement on the region, which is convex or nearly so. If a region is non-convex, the method may perform disastrously, because when locating interior nodes, the Laplacian method is unable to significantly utilize boundary curvature information and boundary nodal point spacing information. Nevertheless, Herrmann [1976] has developed a modification of the technique, by introducing an additional value, usually denoted by w to produce a family of schemes called the Laplacian-Isoparametric schemes, where $0<w<1$. The isoparametric scheme introduced here is different to the scheme introduced by Zienkiewicz & Phillips [1971]. The applicability is restricted to four-sided regions or subregions that are represented by element layout which is transformable to rectangular meshes. This smoothing technique may be useful for modifying the regular grids after it is completely meshed.

5.4 Element Generation Techniques

All the nodal points of the structure found by any of the methods must then be connected to form elements, and the assemblage of the elements forms the continuous structure. The connectivity of the nodes that is required to form an element depends on the type of finite element specified before. For the best results, the shape of the element should be as regular as possible and the number of connections at interior nodes within a region should be as uniform as possible. That is, each node should have approximately the same number of element connections. We are also required to ensure that the boundary nodes and the interior nodes are properly connected for the specific type and number of elements.

Connecting nodes to form elements is a straight forward routine for a regular region, because all the nodal points lie in a straight line. We are required to record all the nodal points into the program, then collect all the necessary nodes for each element as shown in Figure (5.9).

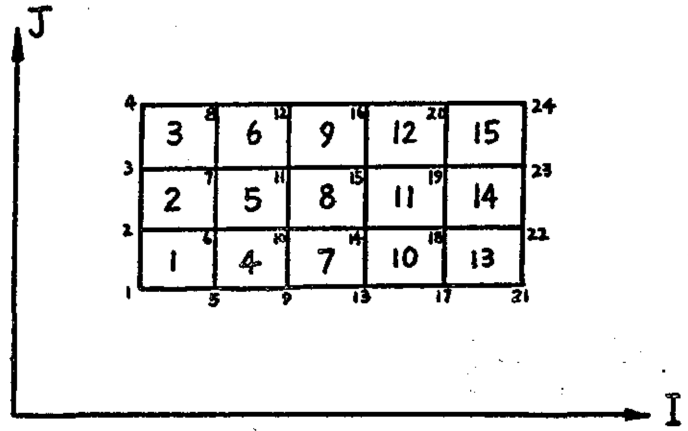


Figure (5.9)

The whole procedure has been fully described in Chenng & Yeo [1979], and also in the review by Bush & Bell [1973], where the method is known as the 'I-J' transformation. However, there are problems involved in connecting nodes to form irregular meshes. The problem is to ensure that all the nodes are properly connected, and that the elements used for finite elements analysis are not too irregular. Frederick, Wong & Edge [1970] have presented a paper to overcome this problem. A special scheme using 'Ghost' points, which are points beyond the region boundary, has been created, eliminates the need to identify external boundary nodes. The technique is used for generating triangular elements. Lewis & Robinson [1978] described a technique of dividing the original data space into disjointed segments. These subsegments are further divided into triangles with no interior nodes, thus forming the elements of triangulation. Denayer [1978] has presented an automatic technique for generating element connectivity. The method involves mapping between an imaginary region, defined by an idealized grid composed entirely of regular regions, and the actual regions to be meshed. The idealized grid is constructed from the boundary curve information, including the number of elements connected to each boundary node.

5.5 Generation of Three-Dimensional Elements

Some of the 2-D mesh generation techniques discussed above, such as Zienkiewicz & Phillips [1971], Gordon & Hall [1973] and Ghessemi [1980], can be extended to three-dimensional problems.

There are papers published which look purely at three-dimensional solid problems. Cook [1974] has described a method involving two types of mesh generators: the surface generator and the volume generator. The method is similar to the transfinite method of Gordon & Hall [1973], which is based on the linearly-blended interpolation formula of Coons [1967] for generating nodal points for general surfaces. The method similarly restricts these curvilinear coordinates (ξ, η) for the surface, generator and (ξ, η, γ) for the volume generator between 0 and 1. The user is required to supply the geometry information from the structure. The method generates interior nodal points inside the surface of volume. However, if it is a surface with many variations of geometry, then input of the nodal points joining each region is required. Also, if there are discontinuous portions of the body, the body is divided into several regions before meshing. However, this method does not give information about connecting nodes to form elements. Frey, Hall & Porching [1979] have introduced an interactive computer program 'PLANIT' in which the intersections of a sequence of planes with the concaternation of 20-node brick elements are interactively constructed. The transfinite interpolation method has been used in coordinate transformation. The method is basically the geometrical verification of the three-dimensional finite element mesh. Nguyen-v-Phai [1982] has presented a mesh generation scheme for tetrahedral elements. The technique, in forming tetrahedral elements is similar to the two-dimensional triangular elements introduced by Frederick et al [1970], which all the connecting lines between two nodal points within the 3-D domain are surrounded in order to form elements. An automatic mesh refinement technique has also been implemented by dividing each original tetrahedral elements into eight smaller elements.

5.6 Using Computer Packages for Finite Element Analysis

Lewis & Cross [1979], Lorensen [1975] and Hoffman [1978] have introduced interactive graphics finite element systems: IFECS, IGFES and IMPRESS respectively. IFECS is based on the technique developed to construct triangular elements, and it is used as an interactive means of interrogating the program output. IGFES is a two dimensional grid generation and it supports all 'NASTRAN' triangular and quadrilateral elements. IMPRESS is a general three-dimensional modelling system in which specific attention is paid to the modelling, using interactive graphics. Fousek [1979] has described an interactive computer graphics system, FAST-DRAW/3 for generation, modification, and display of the finite element models. The technique adopted in this system is based on the 'blending interpolation' equations which were derived by Coons [1967], and utilized by Cook [1974].

Prasad [1979] presented an interface software system, IPAC, which allows a user to have access to other finite element programs. The system is built around 'GIFTS', a modular and interactive graphics oriented finite element program. Three-dimensional isoparametric element generation capability has been introduced for bodies of revolution, such as wheels, torus, rings or similar bodies of irregular cross-sectional features.

5.7 Mesh Generation Algorithms Applied to Other Areas

There are other areas that require the use of the finite element method with mesh generation techniques. Leick & Potvin [1978] have used mesh generation techniques for tubular joint stress analysis. Kalkani [1975] has it used for the 'highway excavation cut' and the method is based on the straight-line interpolation method for node generation and the simple increment method for the element generation. Hakim & King [1978] have applied the method to create a three-dimensional model of the vertebra. The mesh is composed of regularly shaped elements and it avoids extremely skewed elements. Melkes [1978] has also used the technique in magnetic field calculations. This paper is limited only to triangular meshes. Kleinstreur [1980] has introduced interactive mesh generator to simulate flow systems and structural elements. The basic equation uses Galerkin method with weighted residuals. The element generated are triangular for two-dimensional problems.

There are many more interactive graphics finite element systems; many of which can be found in Fredrikson and Mackerle [1980] and Norrie and de Vries [1976].

CONCLUSION

The work presented in this thesis has been devoted to the problem of interfacing the surface description of a car body with the finite element model. It is in the context of working practices at Austin Rover. At the moment, the surface modelling and the finite element modelling are carried out separately which is wasteful of time and money, because two different versions of the geometrical descriptions are produced. Our aim, therefore, has been to cut out the duplication.

The finite element method (FEM) and the shape representation techniques used in CAD have been developed separately but can be unified within the concept of the isoparametric element. Essentially, the element allows any arbitrary geometry to be closely approximated and higher order elements can be used to cope with curved boundaries.

At present, Austin Rover is wanting to develop an integrated system, so that components can be analysed early in the design process. The system will be linked through the 'data-bases', which will contain design, test and analysis information.

The Geometrical Data Base (GDB) which contains the geometry of the car body, is regarded as the master data-base for general purposes. However, we have proposed that the GDB be enhanced to include structural information sufficient to construct the finite element models. A structural Data Base (SDB) has been suggested to include all the geometrical information supplemented by the material properties, including thickness and build information. In addition, a structural consistency check is proposed to ensure that the information in the SDB is correctly defined. It is considered an impossible task to fully automate the checking. Consequently, manual checking, via interactive graphics, has been proposed.

The remaining step is to interpret all the data in the SDB for finite element modelling, which essentially involves meshing individual panels. To mesh, we have to know the type and the number of elements on each panel. But, it is difficult to determine the 'optimum' number of elements and no algorithms have yet been proposed to solve the problem. Generally, programs are very much dependent upon the user's experience and judgement.

The fundamental requirements of automatic mesh generation are an accurate geometry and a mesh control capability. The algorithm must be able to generate elements compatible with adjacent elements and have the appropriate number of nodes according to division parameters, spacing them according to bias parameters.

Basically, the function of the mesh generation algorithm is to generate nodes and form them into elements. It is straightforward to mesh a smooth and regular panel by the simple incremental or 'straight-line' interpolation technique. But the method is inappropriate to irregular shaped panels as arise commonly in internal car structures. We have proposed manual subdivision to overcome the difficulties of automatic meshing. There is no difficulty in implementing manual interfacing as interactive graphic display are commonly available. However, more work is required to find a suitable way of generating nodes on irregular panels.

Usually, the number of elements used for a panel is greater than required. The reason is that the meshing is based on conservative rules of thumb, for example, restrictions on the aspect ratio. However, there are circumstances where long thin elements will not give poor results. Further research is required to resolve the problem.

In future, more work will be necessary to find a way to optimise the choice of element type and the arrangement of elements on each panel. According to theory, the higher the order of the elements and the greater the number the more accurate the solution. However, it will cost more in terms of computer time and storage. Therefore, we have to balance the costs and the accuracy for each individual job.

When assembling all the panels together, we have to ensure that all the connecting boundaries are smooth and without any gaps. One problem is that panels may have various thicknesses, which means that all the 'common' nodes are at different positions. If this occurs, adjustment is needed to ensure that all the boundary nodes are at the same positions.

So far, we have discussed various problems which will cause difficulties in automatic meshing. However, these problems can all be overcome by allowing the user to make the decisions, (probably at a interactive graphic terminal). Even with a large user involvement, there would still be a significant reduction in the time needed to create a finite element model.

In the thesis, we have confined our proposal to the system used at Austin Rover. However, the interfacing of the surface description with the finite element model is a general problem.

Our work indicates the general way forward. It will be necessary to enhance standard formats for the geometry, for example, IGES (Initial Graphics Exchange Specification), which is recognised by the National Bureau of Standards, NBSIR, to include structural information sufficient to construct a finite element model specified to a standard format, for example, the 'Universal' format, which is recognised by SDRC.

REFERENCES

1 Advisory Council for Applied Research and Development (1980),' Aided Design and Manufacture', Her Majesty's Stationary Office.

2 Argyris, J H and Kelsey, S (1960), 'Energy Theorems and Structural Analysis', Butterworth, (reprinted from Aircraft Eng., 1954-55).

3 Ball, A A (1974), 'CONSURF. Part 1: Introduction of the Conic Lofting Tile', Computer Aided Design, 6, 4, 243-249.

4 Ball, A A (1975), 'CONSURF. Part 2 : Description of the Algorithms', Computer Aided Design, 7 , 4, 237-242.

5 Ball, A A (1978), 'CONSURF. Part 3 : How the Program is used', Computer Aided Design, 9, 1, 9-12.

6 Bartz, J A (1971), 'A 3-D Computer Simulation of a Motor Vehicle Crash Victim', Calspan Corporation report VJ-2978-V-1.

7 Bates, K J (1972), 'The Autoken Automative and Aerospace Packages', Proc. Curved Surfaces in Engineering Conf., Camb., 19-22.

8 Bezier, P (1968), 'Systeme UNISURF', Automatismes, 13,5.

9 Buell, W R and Bush, B A (1973), 'Mesh Generation Survey', Trans. ASME, J. Eng Ind. 332-338.

10 Butlin, G A (1983) 'Interfacing C.A.D. and F.E.M.', Effective CADCAM, (Wadsworth, D S, ed) CAMB.

11 Cheung, Y K and Yeo, (1979), 'A Practical Introduction to Finite Element Analysis', Pitman Publishing Limited.

12 Clough, R W (1960), 'The Finite Element in Plane Stress Analysis', Proc. 2nd A.S.C.E. Conf. on Electronic Computation, Pittsburgh, Pa. 345-78.

13 Cook, W A (1974), 'Body Oriented (Natural) Co-ordinates for Generating Three-dimensional meshes', Int. I. Num. Meth. Engng, 5, 27-43.

14 Coons, S A (1967), 'Surfaces for Computer-Aided Design of Space Form', Project Mac, MIT.

15 Davy, D W (1972), 'Surface Manipulation and Press Tool Engineering in Car Body Maintenance', Curved Surfaces in Engineering, 63-69.

16 Deane, J (1982), 'Design Software', CADCAM Int., 51-53.

17 Denayer, A (1978), 'Automatic Generation of Finite Element Meshes', Comp. Struct, 9, 359-364.

18 Durocher, L L and Gasper, A (1979), 'A Versatile Two-Dimensional Mesh Generator with Automatic Bandwidth Reduction', Comp. Struct, 10, 561-575.

19 Egeland, O and Araldsen, P O (1974), 'SESAM-69-General Purpose Finite Element Program", Comp. Struct, 4, 41-68.

20 Emmerson, W C and Fowler, J E (1974), 'The Application of Computer Simulation in Vehicle Safety', Proc of 5th Int. Conf. on Experimental Safety Vehicles, U.S. Dept., of Trans., London, 712-720.

21 Emmerson, W C (1976), 'CAD in the Motor Industry Leyland Cars', CAD, 8, 193-197.

22 Faux, I D and Pratt, J M (1979), 'Computational Geometry for Design and Manufacture', John Wiley & Son.

23 Ferguson, J C (1964), 'Multivariable Curve Interpolation', J.ACM, 11, 2, 221-228.

24 Flutter, A G and Rolph, R N (1976), 'POLYSURF : An Interactive System for Computer-Aided Design and Manufacture of Components', CAD 76 Proc., 150-158; CAD Centre, CAMB.

25 Forrest, A R (1970), 'Coons Surfaces and Multivariable Functional Interpolation', Univ. of Camb., Computer -Aided Group, Document 38.

26 Forrest, A R (1972), 'Mathematical Principles for Curve and Surface Representation', Curved Surface in Engineering, 5-13.

27 Fousek, D P (1979), 'FASTDRAW/3 : Interactive Graphics Mesh Generation', Conf. Electronic Computation, 7th Washington Univ., St. Louis, Mo, ASCE, N.Y. 716-730.

28 Frederick, O C, Wong, Y C and Edge, F W (1970), 'Two-Dimensional Automatic Mesh Generation for Structural Analysis', Int.J.Num.Meth.Engng, 2, 133-144.

29 Fredrickson, B and Mackerle, J (1980), 'Structural Mechanics Finite Element Computer Programs : Survey and Availability', Rep. Lithikp-R-054, Linkopling Inst. of Tech., Sweden.

30 Frey, A E, Hall, C A and Porsching, T A, 'An Application of Computer Graphics to Three-Dimensional Finite Element Analysis', Comp. Struct, 10, 149-154.

31 Fujii, T and Yuki T (1973), 'Automatic Data Generation and Large Matrix Operation for Three-Dimensional Plate Structure Analysis', Theory and Practice in Finite Element Analysis, Univ. of Tokyo Press.

32 Galerkin, B G (1915), 'Series Solution of Some Problems of Elastic Equilibrium of Rods and Plates', (Russian), Vestn. Inzh. Tech, 19, 897-908.

33 Ghassemi, F (1982), 'Automatic Mesh Generation Scheme for a Two-or-Three-Dimensional Triangular Curved Surface', Comp. Struct., 15, 6, 613-626.

34 Giles, J G (1971), 'Body Construction and Design'; Auto-Tech Series, 6.

35 Gordon, W J (1971), 'Blending Function Methods of Bivariate and Multivariate Interpolation and Approximation', SIAM J.NUM.ANAL, 8, 1.

36 Gordon, W J and C A Hall (1973), 'Construction of Curvilinear Co-ordinate Systems and Applications to Mesh Generation', Int.J.Num. Meth. Engng, 7, 461-477.

37 Guyan, R J (1965), 'Reduction of Stiffness and Mass Matrices' AIAA J. 3, 2.

38 Hakion, N S and King, A I (1978), 'A Computer-Aided Technique for the Generation of a 3-D Finite Element Model of a Vertebra', Comp. Biol. Med., 8, 187-196.

39 Haugerud, M H (1978), 'Interactive 3-D Mesh Generation by an Idealization and Mapping Technique', 3rd Int. Conf. and Exh. on Comp. in Eng. and Building Design, 14-16, 753-66.

40 Heber, R, Shephard, M S, et al., (1981), 'A general two-dimensional, graphical finite element preprocessor utilizing discrete transfinite mappings', Int. J.Num. Meth. Engng., 17, 1015-1044.

41 Herrmann, L R (1976), 'Laplacian-Isoparametric Grid Generation Scheme', J. of the Eng. Mesh. Div., ASCE, 102, 749-756.

42 Hoffman, R E (1978), 'Interactive Graphics Finite Element Mesh Generation', Conf. on Comp. in Civ. 1st Pap. Atlanta, 700-713.

43 Imafuku, I and Kodera, Y (1980), 'A Generalized Automatic Mesh Generation Scheme for Finite Element Methods', Int.J. Num. Meth. Engng., 15, 713-731.

44 Irons, B M (1966), 'Engineering Application of Numerical Intergation in Stiffness Method', AIAA J, 14, 2035-7.

45 Jones, R E (1974), 'A Self-Organizing Mesh Generation Program', J. of Pressure Vessel Tech., 96, 193-199.

46 Kalkani, E C (1974), 'Mesh Generation Program for Highway Excavation Cuts', Int. J. Num. Meth. Engng, 8, 369-394.

47 Kamel, H A and McCabe, M W (1973), 'A graphics Oriented Interactive Finite Element Time Sharing Package, Gifts' Research Report, Univ. of Arizona, Tucson.

48 Leick, R D and Potvin, A B (1977), ' Automated Mesh Generation for Tubular Joint Stress Analysis', Comp. Struct, 7, 73-91.

49 Lewis, B A and Robinson, J S (1978), 'Triangulation of Planar Regions with Applications', Comp. J, 21, 324-332.

50 Lewis, B A and Cross, M (1978), 'IFECS an Interactive Finite Element Computing System', Applied Math. Modelling, 2, 165-176.

51 Pian, T H H (1964), 'Derivation of element stiffness matrices by assumed stress distribution', AIAA J, 2, 7 1333-1336.

52 Liming, R A (1944) 'Practical analytic geometry with applications to Aircraft', MacMillian, N.Y.

53 Lipchin, L and Little, A D (1982), 'Managerial and Strategic Planning Aspects of CAD/CAM technology implementation', CAD 82, 5th Int. Conf. and Exh. on Comp. in Design Engineering, (Pipes, A, ed), Butterworths, Sussex.

54 Lorenson, W E (1974), 'Interactive Graphics Finite Element System (IGFES)', Conf. Struct. Mech. Comp. Prog., Univ. of Maryland, 991-1001.

55 MacNeal, R H (ed) (1976), 'The NASTRAN Theoretical Manual', NASA SP-221(03).

56 MacNeal, R H and McCormick, C W (1976), 'The NASTRAN Computer Program for Structural Analysis'.

57 Melkes, F (1978), 'Automatic Mesh Generation in Magnetic Field Calculations', Comp. 78-Conf. on the Comp. of Magnetic Field, Grenoble, 11.2/1-8.

58 Moss, S P (1979), 'Use of the Finite Element in the Motor Industry at Pressed Steel Fisher, Finite Element News.

59 Nicoles, V T and Citipitiogm, E (1978), 'SDRC-SUPERB', Structural Dynamics Research Corporation.

60 Norrie, D H and de Vries, G (1976), 'A Finite Element Bibliography', Plenuj Press, N.Y.

61 PAFEC Ltd (1978), 'PAFEC 75', PAFEC House, Nottingham.

62 Phai, V N (1982), 'Automatic Mesh Generation with Tetra-hedron Elements', Int.J.Num.Meth. Engng, 18, 273-289.

63 Prasad, B (1979), 'IPAC-an Interface Finite Element Method Model Generation Package', Conf. Elect. Comp. 7th Washington Univ. ASCE, N.Y., 829-844.

64 Przemieniecki, J S (1968), 'Theory of Matrix Structural Analysis', McGraw-Hill, N.Y.

65 Ness, P H (1983), ' Managing Integrated CAD/CAM in a Large Aerospace Company for Fun and Profit', Auto. Tech. for Man and Prod. Adv. Through CAD/CAM and Engineering Data Handling, 181-189, (Wang, P.C)

66 Sabin, M A (1972), 'Numerical Master Geometry', Proc. Curved Surface in Engineering Conf., CAMB, 23-25.

67 Sabin, M A (1971), 'An Existing System in the Aircraft Industry', The British Aircraft Corporation Numerical Master Geometry System, Proc. Roy. Soc. Lond., A321, 197-205.

68 Schrem, E and Roy, J R (1971), 'An Automatic System for Kinematic Analysis ASKA, Part 1', Proc. of the IUTAM Symp. on High Speeding Comp. of Elastic Struct., 2, 477-507.

69 Sagawa, K (1973), 'Automatic Mesh Generation for Three-Dimensional Structural Based on Their Three Views', Theory and Prac. in Finite Element Struct. Analysis, Univ. of Tokyo Press.

70 Singh, S (1976), 'A Computer-Aided Finite Element Idealization and Mesh Generation System', CAD 76, 2nd Int. Conf. Comp. in Eng. and Building Design, Imperial College, IPC Sci. and Tech. Press, 213-218.

71 Stefanou, G D (1980), 'Automatic Triangular Mesh Generation in Flat Plates for Finite Element', Comp. Struct., 11, 439-464.

72 Suhara, J and Fukuda, J (1972), 'Automatic Mesh Generation for Finite Element Analysis', Adv. in Comp. Meth. in Struct. Mech. and Design, 607-624.

73 Szabo, B A and Lee, G C (1969), 'Derivation of Stiffness Matrices for Problems in Elasticity by Galerkin Method', Int. J. Num. Meth. Engng, 1, 301-310.

74 Thacker, W C (1980), 'A Brief Review of Techniques for Generating Irregular Computational Grids', Int. J. Num. Meth. Engng, 15, 1335-1341.

75 Turner, M J, Clough, R W, Martin, H C and Topp, L J (1956), 'Stiffness and Deflection Analysis of Complex Structures', J. Aero. Sci., 23, 805-23.

76 Welbourn, D B (1981), 'The Practical Use of Computers in Pattern, Mould and Die Making', Wofson Camb. Industrial Unit, Camb.

77 Wu, E R (1980), 'Some Finding in Using the Program, Versatile 2-D Mesh Generator with Automatic Bandwidth Reduction', Comp. Struct, 12, 181-183.

78 Zienkiewicz, O C and Phillips, D V (1971), 'An Automatic Mesh Generation Scheme for Plane and Curved Surface by 'Isoparametric' Coordinates', Int. Num. Meth. Engng, 3, 519-528.

79 Zienkiewicz, O C and Cheung, Y K (1967), 'The Finite Element in Structural and Continuum Mechanics', McGraw-Hill, London.

80 Zienkiewicz, O C and Cheung, Y K (1965), 'Finite Elements in the Solution of Field Problems', The Engineer, 507-510.

81 Zienkiewicz, O C (1971), 'The Finite Element Method in Engineering Science', McGraw-Hill, N.Y.

82 Zienkiewicz, O C (1979), 'The Finite Element Method', 3rd Edition, McGraw-Hill, UK.

83 Kleinstreuer, C and Holdeman, J.T (1980), 'A Triangular Finite Element Mesh Generator For Fluid Dynamic Systems of Arbitrary Geometry', Int. Num. meth. Engng, 15, 1325-1334.