

DATA PREPARATION SYSTEM FOR FINITE ELEMENT
STRESS ANALYSIS IN THREE DIMENSIONS

by

D T ETELA
B.Sc.(Eng) , A.C.G.I

October 1981

A thesis submitted for the degree of Doctor of Philosophy of the University
of London and for the Diploma of Imperial College

Mechanical Engineering Department
Imperial College of Science &
Technology
LONDON SW7

ABSTRACT

Engineering design of various forms go through an iterative loop of stress and cost analysis, to arrive at an optimum solution of the design problem. Due to the fast growth and falling cost of digital computers, numerical methods for solving stress problems have become very popular for most problems when an analytical solution is very difficult or virtually impossible. The Finite Element method being the most popular of these numerical techniques for stress analysis, has an inherent problem of requiring a large amount of time in preparing and checking the geometric input data.

Under a Computer Aided Design of steel and concrete structure with reference to nuclear power stations, data preparation for structural finite element analysis of two-dimensional and three-dimensional problems is considered in this work. The mesh generating package produced is linked to an existing general draughting system which has been enhanced to cope with the requirements of the package. Most of the main programs have been written in standard Fortran for machine transferability. The programs are written to interact with, and guide the user through a problem. With little modification the package can be used on a mainframe computer in batch mode.

ACKNOWLEDGEMENT

Special thanks are due to the government of the Rivers State of Nigeria and the Nigerian Army for sponsoring this work at Imperial College of Science and Technology, London.

With the deepest appreciation, I acknowledge the counsel, guidance, suggestions and encouragement of Dr C B Besant my supervisor, and late Charles Hsiung without whose help and enthusiasm this project would never have been successful.

Thanks are due to the following wonderful people for all their help and support: T Eagles, L Eggleston, D Thompson, C Lui, G Stavrides and S Khurmi. Thanks are also due to Miss V Trout for typing the thesis, J Wynne and I Armstrong of Rolls Royce Ltd Bristol for their help in the case study and the latter for his advice on the formulation of the loading ratios.

Finally, I am indebted to the following for their help and advice in my use of FINEL: D Hitchings, F Maali and K Balasubramaniam.

D T ETELA

C O N T E N T S

Page

ABSTRACT

CHAPTER 1	INTRODUCTION	1
1.1	DESIGN METHODS	3
1.1.1	The iterative method	3
1.1.2	The direct design method	5
1.1.3	The design selection method	5
1.2	COMPUTER AIDED DESIGN (CAD)	6
1.3	FINITE ELEMENT METHOD (FEM)	7
1.4	CAD A TOOL FOR FEM	12
CHAPTER 2	COMPUTER AIDED DESIGN SYSTEM HARDWARE	15
2.1	COMPUTER HARDWARE	18
2.1.1	Mainframe based system	20
2.2	THE SAMG SYSTEM HARDWARE	21
CHAPTER 3	CAD SYSTEM SOFTWARE	30
3.1	INTRODUCTION	30
3.1.1	The operating system	30
3.1.2	Scope of SAMG	32
3.2	THE GRAPHICS SYSTEM	34
3.2.1	The menu	37
3.2.2	System modules	38
3.2.3	Data Storage and File Handling	40
3.2.4	Displaying, Plotting and Windowing the Workspace	45
3.3	APPLICATION PROGRAMS	47
3.3.1	The use of the system for draughting	48
3.3.2	The use of the system for animation	49
CHPATER 4	AUTOMATIC MESH GENERATION FOR 2D AND AXISYMMETRIC SHAPES	53
4.1	THEORY	54
4.1.1	Coordinate Geometric Algorithm	63
4.1.2	Mesh Concentration	68
4.2	JOINING THE QUADRILATERALS	70
4.3	MESH DATA STORAGE AND FILE MANIPULATION	76
4.4	2D-MESH EDITING AND AUTOMATIC RENUMBERING OF NODES	80

	Page
4.4.1 Add Element	80
4.4.2 Delete Element	81
4.4.3 Adjust position of nodes	82
4.4.4 Automatic renumbering	82
4.5 DOUBLE AND ROTATE	83
4.5.1 Theory	84
4.5.2 Nodal Point Coordinates and Element Connection for 2½D Files	87
CHAPTER 5 3-D SHAPE DESCRIPTION AND MESH GENERATION	89
5.1 INPUT METHODS AND THE MACROBLOCKS	90
5.2 MESHING THEORIES	97
5.2.1 Coordinate Geometry Algorithm 3-D	98
5.2.2 Isoparametric Mapping	102
5.2.2.1 Node numbering and computing scaled coordinates	105
5.2.3 Matrix transformation	111
5.2.3.1 Scaling	112
5.2.3.2 Translation	112
5.2.3.3 Rotation	112
5.3 JOINING BUILDING BLOCKS	114
CHAPTER 6 DATA HANDLING FOR GRAPHICS AND 3-D MESH	119
6.1 DISPLAYING AND PLOTTING 3-D GRAPHICS DATA	120
6.1.1 Plotter Character Generation	129
6.2 3-D MESH DATA	131
CHAPTER 7 CASE STUDY	141
7.1 PROBLEM SPECIFICATION	141
7.2 SOLUTION	144
7.2.1 Use of FINEL	145
7.3 RESULTS AND COMMENTS	146
CHAPTER 8 CONCLUSION AND FUTURE DEVELOPMENT	152
REFERENCES	156
APPENDIX A	159
APPENDIX B	169
APPENDIX C	177

C H A P T E R 1

INTRODUCTION

The successful engineering product is the wake or result of weeks or months of hard work by one or more design engineers. From the smallest dentist's drill to the largest multimillion oil refinery, the design engineer goes through a standard set of procedures ranging from the initial sketch to the final design drawings guided by his training, experience, intuition, and the objective of the design. Another important factor that must be considered by the engineer apart from his design criteria is the possibility of producing what has been designed, since it is of no use in having a fantastic design satisfying all other design criteria, that cannot be produced practically and economically. From the beginning to the end of a particular design problem, the acceptability of the design is certified by series of calculations formulated by the engineer, using the boundary conditions imposed upon the design by the objective. The formulation, solution, and analysis of the results of the calculations are integral parts of the Design Methods, and can be time consuming.

In the past where direct solutions for some design problems have not been possible, various techniques were used in obtaining approximate solutions, or models were made when practical, and tested to their limits. Highly skilled craftsmen (not very many these days) were employed to produce scaled-down models of large designs and various tests were performed

on the models to aid the designer predict the behaviour of the full scale product. Sometimes the formulated problems were highly simplified and solved, and different criteria were used in computing safety-factors to offset the simplification. One of the standard safety-factor computations led to the thickening of stress bearing members in a structure, but this cannot be accepted in places such as the modern aircraft industries where weight plays an important role, hence demanding more realistic and accurate solutions.

Numerical solutions which are fundamentally more tedious for the designer can be used where analytical solutions fail. With the advent of the digital computer, the past two and half decades have seen a tremendous improvement in the solution of various kinds of problems using numerical methods. A host of these methods have been developed to a high standard within this period. The success of Finite Element Method (a numerical technique) for the dynamic and static analysis of general structure and transport problems cannot be overemphasised. For at present there are numerous computer program packages for finite element analysis, and unless for teaching, research, or specialization purposes, it will be uneconomical to develop a finite element program.

The designer now armed with the digital computer and speedy solution techniques can now look through his design methods and usefully employ the machine and all its advantages in all possible facets of his manifold task.

1.1 DESIGN METHODS

To design means to determine the essential properties and features of the object in question, and contrive a solution, for the solution of an engineering problem. The object in question might have different properties, which may be classified under different aspects. For instance when a customer wants to buy a machine, he has in mind some specific properties of input and output of his machine. The engineer then describes the function of the machine relying on three fundamental methods of design, any of which may be used in varying degree on the particular design process.

- a) The iterative method
- b) The direct design method
- c) The design selection method

1.1.1 The iterative method

This is the most frequently used method, especially in the early stages of design. The process consists of employing intuition and experience, from which a set of rules, drawn from experience or direct design, is used to produce a preliminary solution. The resulting design is then analysed to determine if it meets the specified design requirements. Should it fail to do so, it is then modified on the basis of further information made available from the analysis, and re-analysed until it either meets the specification or proved not feasible within the design constraints, Fig 1.

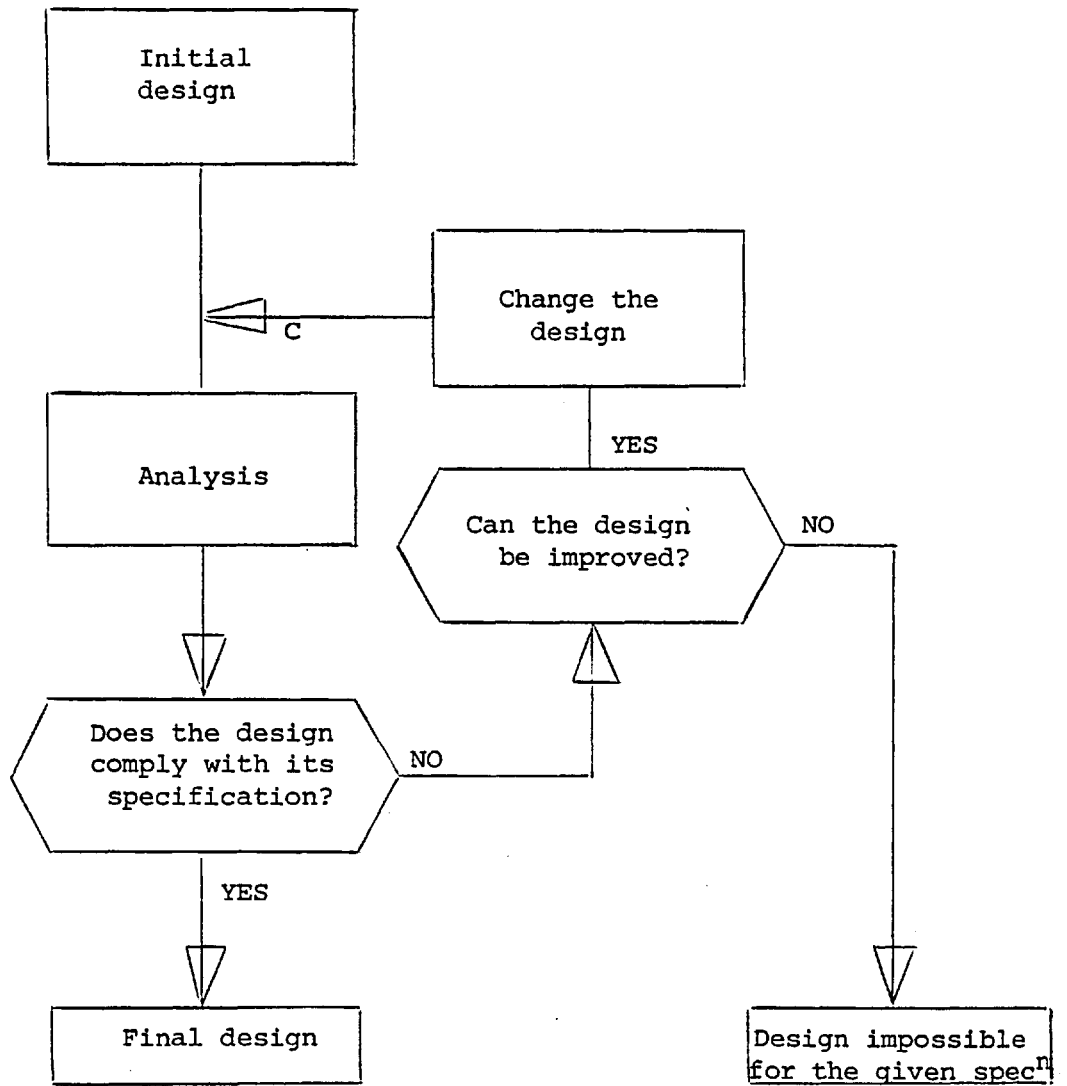


Fig. 1 The iterative design method

1.1.2 The direct design method

Whereas the iterative method uses the results of a design analysis as a basis for re-design decision, the direct design method uses the design analysis as a method for affecting the design. In other words a design is produced directly by analysis. This process implies that a design can be defined by a parametric representation, and that the interactions between parameters are known.

1.1.3 The design selection method

A design process may yield a number of solutions which all satisfy given constraints. This is applicable to the situation where the design constraints themselves have one or more variables. In such a case the selection of a final design given the performance characteristics of each design is largely a matter of human judgement by compromising performance as it is affected by design constraints.

Whatever the design method selected, the whole design process goes through five stages of: Specification, Strategy, Solution, Analysis and Application. A detailed description of these stages of design can be obtained from Besant [1] . The use of the digital computer in these separate stages of design can only be seen as aiding the designer with the machine.

1.2 COMPUTER AIDED DESIGN (CAD)

Whereas most of the work of the designer is concerned with the production of good quality drawings, schedules and other documentation geared towards production, the design requires calculations relating to stress analysis, heat transfer, and costing but to mention a few. In the design process, these calculations form an integral part for deciding whether the design meets the required specifications. These calculations can prove to be labourous and subject to considerable error if performed by hand. Large computers have been used in helping the designer in the calculations, since the machines were primarily designed for faster calculation. It was quite desirable to support all the manifold activities of the designer by the computer and not just the calculations. Computer graphics was not viable with the early generation of computers which were simply christened "Number Crunchers" [2]. With the development of digital computers, interactive computer graphics became feasible in the late 1950's.

The single event that did most to promote interactive computer graphics as an important new field was the publication in 1962 of a thesis "SKETCHPAD : A man-machine graphical communication system" by Sutherland [3]. The thesis proved to many readers that interactive computer graphics was a viable, useful and exiting field of research. Computer graphics has taken a long stride since then. Draughting with the computer is now a common practice with successful large engineering firms.

The computer could then be economically employed to improve or speed up different aspects of the design functions, dealing with the more repetitive and tedious types of work in the design process, hence the birth of CAD.

CAD has developed into a process where man and machine are blended into a problem solving team, efficiently utilizing the best characteristics of each part of the team. Implying that the computer is not used when the designer (man) is more effective such as intuition, judgement, experience and imagination. Similarly not using man when the machine is more effective such as storage and recovery of detailed information, and tolerance of repetitive work. It is therefore the complementary marriage of the better characteristics of man and machine which is the most important aspect of CAD, Fig I.2. The limitation of their problem solving capacity becomes functional to that of man, computer, and their interfacing software and peripheral devices.

1.3 FINITE ELEMENT METHOD (FEM)

The finite element method is now a well established numerical technique in the static and dynamic analysis of general structures and the solution of continuum problems. A detailed approach to the subject can be obtained from Zienkiewicz [4].

The theory is based on an elastic structure or continuum being represented by discrete components or elements

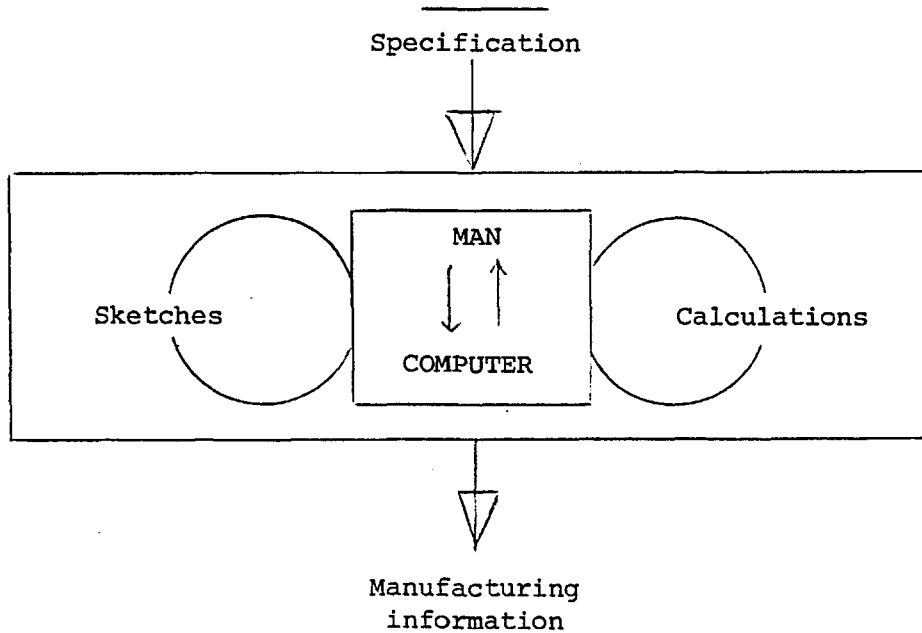
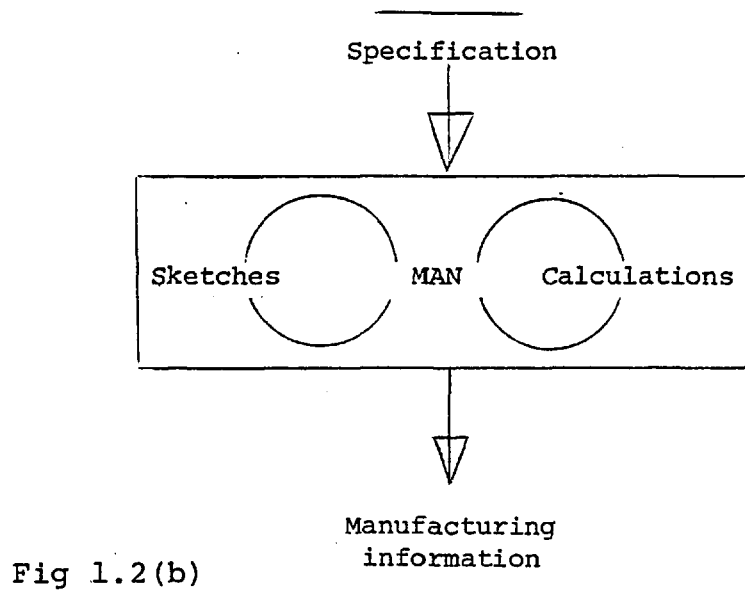
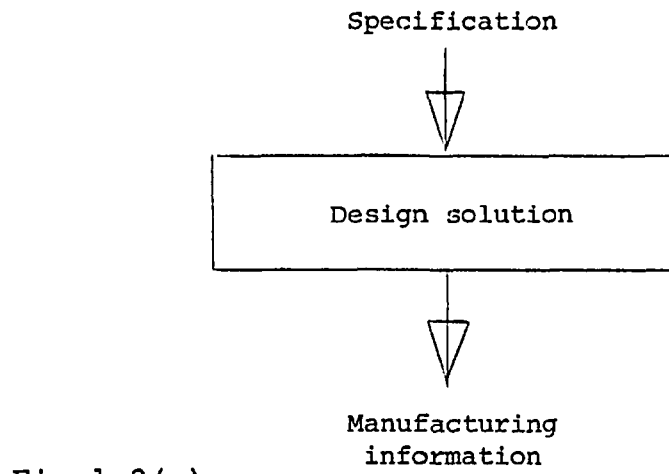


Fig 1.2(c) The Design Process for CAD

interconnected at a finite number of points situated on the element boundaries. The displacements of these nodal points being the basic unknown parameters of the problems.

A set of functions is chosen to uniquely define the state of displacement within each finite element in terms of its nodal displacements, thus defining the state of strain within an element. These strains with any initial strain and the constituent properties of the material, would then define the state of stress throughout the element. A system of forces equilibrating the boundary stresses and distributed loads is determined, resulting in a stiffness relationship of the form:

$$\{F\}^a = [K]^a \{\delta\}^a + \{F\}_p^a + \{F\}_{e\phi}^a \quad 1.1$$

where

$\{F\}^a$ represents the nodal forces on an element 'a'

$[K]^a$ is the element stiffness matrix

$\{\delta\}^a$ the nodal displacement of the element 'a'

$\{F\}_p^a$ the nodal forces required to balance any distributed loads acting on the element

$\{F\}_{e\phi}^a$ the nodal forces required to balance any force due to strains such as may be caused by temperature change, if nodes are not subject to any displacement

Similarly stresses or internal reactions can be specified at any point of the element in terms of nodal displacements:

$$\{\sigma\}^a = [S]^a \{\delta\}^a + \{\sigma\}_p^a + \{\sigma\}_{e\phi}^a \quad 1.2$$

where

$\{\sigma\}^a$ represents the nodal stresses of the element 'a'

$[S]^a$ the stress matrix

and the last two terms are stresses due to distributed load on the element boundaries, and the initial stresses respectively when no nodal displacements occur.

The choice of a displacement function to satisfy the requirements of displacement continuity is not an easy one, and could invalidate compatibility conditions if not defined correctly. Concentrating equivalent forces at the nodes can only satisfy the equilibrium conditions in the overall sense for the element, but local violation of equilibrium within each element will always occur. This will at first sight suggest that the more the number of elements or nodes per element, the greater the accuracy of the modelling. But then the greater will be the number of equations to solve, hence the cost of the solutions. Somehow a balance has got to be struck. The choice of element shape and the form of the displacement functions will generally determine the accuracy of the finite element model. Elements can take a variety of forms, the simplest being beams, triangles and quadrilaterals. The type of element in conjunction with the nodal forces and displacements, determine the stiffness matrix of the element in equation 1.1.

There are many computer programs (NASTRAN, ASKA, BERSAFE, PAFEC, FINEL etc) written to produce the stiffness matrix of each element, assemble the individual stiffness matrices to form the overall stiffness matrix, insert the appropriate boundary condition and solve the resulting system of matrix equations. Hence obtaining the deflections and stresses throughout the structure. The solution technique of all the programs might differ slightly from each other, but common to all, the input data of finite element programs consist of geometric idealization, material properties, and the loading and boundary conditions. The major problem, and a large portion of the input data is in the geometric representation of the structure by suitable mesh, [1].

The great freedom given to the users of the finite element method to pursue more fundamental problems is simultaneously restricted by the sheer amount of input demanded of them for realistic applications¹¹. Simply, the amount of time needed to prepare data for the solution of real problems can run into weeks, and in certain difficult cases, months. Manually prepared, these data are bound to return for corrections of either a minor nature such as mere typing error in the preparation of input cards, or a major type such as incorrect geometric coordinate data obtained during calculation. Some form of automation would be welcome to ease data preparation for the finite element analysis.

1.4 CAD A TOOL FOR FEM

The idea of automatic mesh generation is neither novel nor uncommon, for as far back as 1956, mesh generation programs for simple constant stress triangles were written by Turner et al [5] , in two dimensional plane. For more complicated two-dimensional shapes, various methods have been developed and to mention but one, Crisp [6] in 1969, developed a mesh generation program to fit triangular elements in a uniform or non-uniform manner in a primitive block within a structure. These primitive blocks were then connected at their boundaries manually.

One of the earlier three-dimensional mesh generation programs was written by Latham [7] for cylindrical shell type structures with only one element through the thickness, and cylinder to cylinder intersection was analysed by manual preparations. Difficulties in defining complicated three-dimensional shapes to the digital computer remained a problem to the automatic mesh generation drive. In 1972 Neale [8] used Hexatredral super-elements or macroblocks which are in turn joined at their boundaries. The use of isoparametric mapping given by Ergatoudis, et al [9] , gave the latter a good foundation in modelling curve-surfaced macroblocks, but the limitation of using isoparametric mapping (ie, the inability to exactly model certain curves like circles) remained.

The developments in CAD have been orientated mainly towards interactive graphics and simulation. Since it is possible to produce meaningful drawings with the graphics

terminal by defining the boundaries of an object in a digital format, it is only logical to assume that with little effort, the geometric coordinates of any point within a given structure that is drawn can be obtained. The writing of computer programs to generate the coordinates of mesh nodes, element connection of the meshes and the topology of the elements have been popular for fairly trivial cases, mainly in two-dimensional analysis. For three-dimensional analysis such programs have been restricted to cylinders and beams or axisymmetric problems. There are few commercial packages for three dimensional mesh generation that are run on large computers which are expensive to hire or buy and are also expensive to run.

A low cost, and efficient general program for developing the meshes, computing the coordinates of the nodes and preparing the topology data file can only be seen in the light of interactive computer graphics, where errors can easily be detected and corrected by the operator. This is evident because, visual assimilation of the shapes of objects is far more efficient and quicker than going through a long list of numbers representing the geometric coordinates of the boundaries of the objects. Once the meshes are created, and the shape thus produced on a visual display unit by the assembly of these meshes is satisfactory, the coordinates of the nodes on each element and the topology can be stored in permanent files, eliminating typing and other minor human errors. Thence it can be seen what an attractive tool CAD is for FEM.

CAD has been successfully used in the preparation of data for complicated plane structures and axisymmetric shapes [10] for finite element analysis. In some cases the time required to prepare the data by hand was reduced by more than 90% using the CAD technique. In the same work [10], CAD was also used in analysing the results obtained in the finite element solution, displaying nodal deflection and stresses graphically.

The desirability of CAD as a tool for FEM is thus multifolded, the time reduction, the graphical representation as a post and pre-processor, the ease of usage due to its interactive nature, and the cost reduction if a minicomputer or a microprocessor is used.

C H A P T E R 2

COMPUTER AIDED DESIGN SYSTEM-HARDWARE

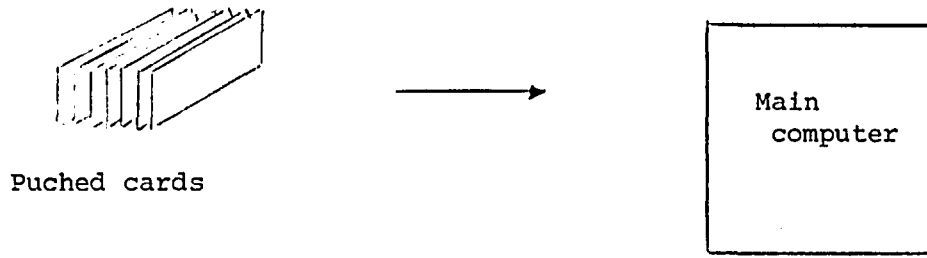
The computer can now be used in a number of ways to aid the designer in almost all stages of the design process, and a number of techniques are available for communication between man and machine, each of which has different characteristics. The technique selected by the designer will depend on: the design stage, system hardware, system software, and funds allocated for the design. There are four main levels of communication between man and machine, Fig 2.1:

a) Batch Mode

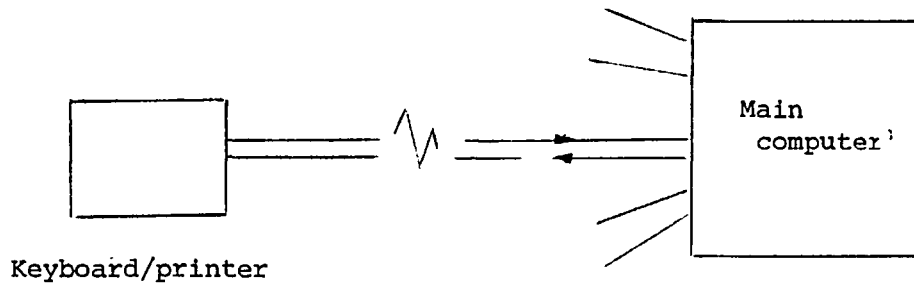
The simplest method of computing is the batch mode where the user prepares data and operating instructions on a suitable storage medium, reads them into the machine, and later receives a print out of results and comments from the machine. The results can be presented by the machine in the form of numbers or partially processed as graphs or drawings.

b) Remote Terminal

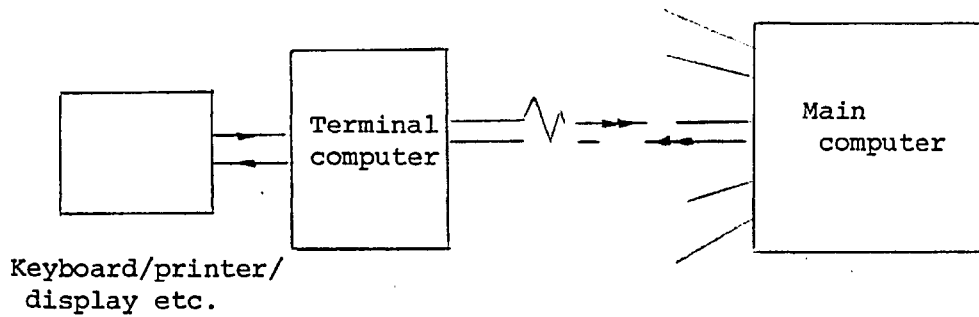
The use of a computer via a G.P.O. telephone line from a remote terminal such as simple teletype keyboard and a line printer or a more complex keyboard with graphic display and hard copy facilities. In this mode of communication, there is a degree of interaction between the user and the machine. Data or instruction can be supplied and received to and from a running program.



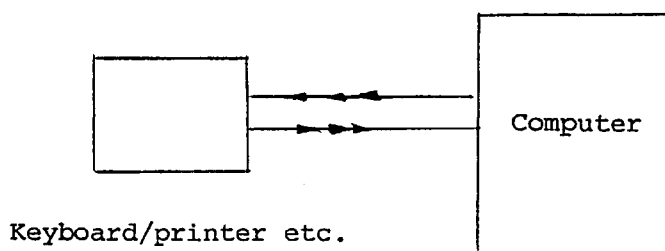
Batch processing



Remote Terminal processing



Intelligent terminal processing



On-line processing

Fig 2.1 Processing Techniques

c) Intelligent Terminal

Normally consists of a small computer (minicomputer or micro-processor) with one or more peripheral devices, connected via a high speed line to a central large computer. This kind of terminal has all the advantages of a remote terminal, and more in the sense that dependency on the main-frame computer for minor data handling and arithmetic processes is eliminated, since the terminal computer can handle such operations. The terminal can have different peripherals like digitising table, card reader, magnetic tape units etc connected to it. With the appropriate supporting system software, the intelligent terminal can be used in a multiuser environment.

d) On-line mode

In this mode of operation, the user communicates directly with the computer through its peripheral devices and can use all available devices to their best advantage. On-line processing is highly desirable if the degree of interaction required is high like in the control of a processing plant where the results of analyses are fed back into the computer which in turn modifies the process to the correct sequence should the result vary from what is expected [12] .

In the work described in this thesis, the programs are written to operate in an on-line mode, and with little modification the package can operate in any of the other levels of communication. The choice of this level of communication is not random but was dictated by the computer hardware (the equipment) used for the research.

2.1 COMPUTER HARDWARE

While it is often convenient to use the term computer to describe the equipment used in processing a program, it is not a single machine in most modern installations, but a series of machines performing different specific tasks [13] . Computers vary in size from desk-top machines to large installations capable of handling many programs and communicating with a large number of remote terminals simultaneously. From an elementary point of view, almost every general purpose digital computer can be described as having the same basic structure, Fig 2.2.

The Central Processor Unit (CPU) could be described as the heart of the computer, consisting of three main sections:

- a) The Arithmetic Unit which performs various numerical manipulations specified by the program, constantly communicating with the various storage devices. For the arithmetic unit to successfully perform its required task, it must be told what to do, hence the Controller.
- b) The Controller which sequentially examines each instruction and directs the action of the computer. The instructions that will be examined by the Controller, program data for immediate use, and intermediate results of mathematical operations from the arithmetic unit would have to be made available in an easy and fast recovery depository. This requirement is met by the final part of the Central Processor known as the Core Store.

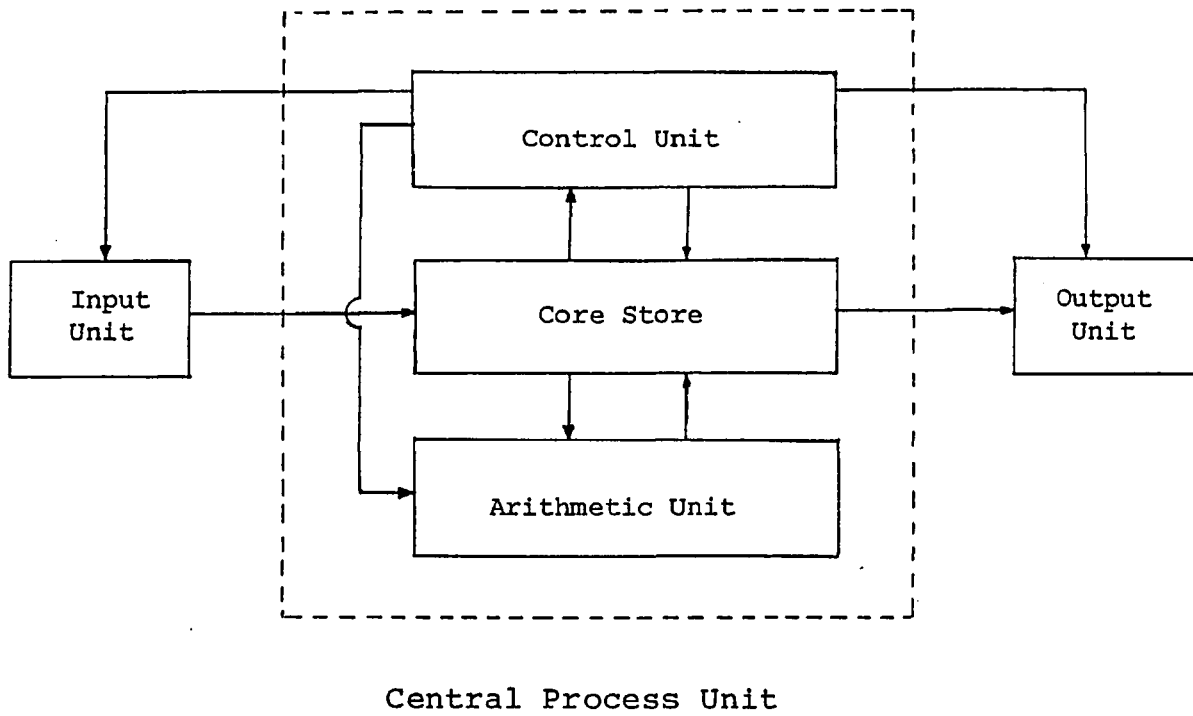


Fig 2.2 General Organisation of a Computer

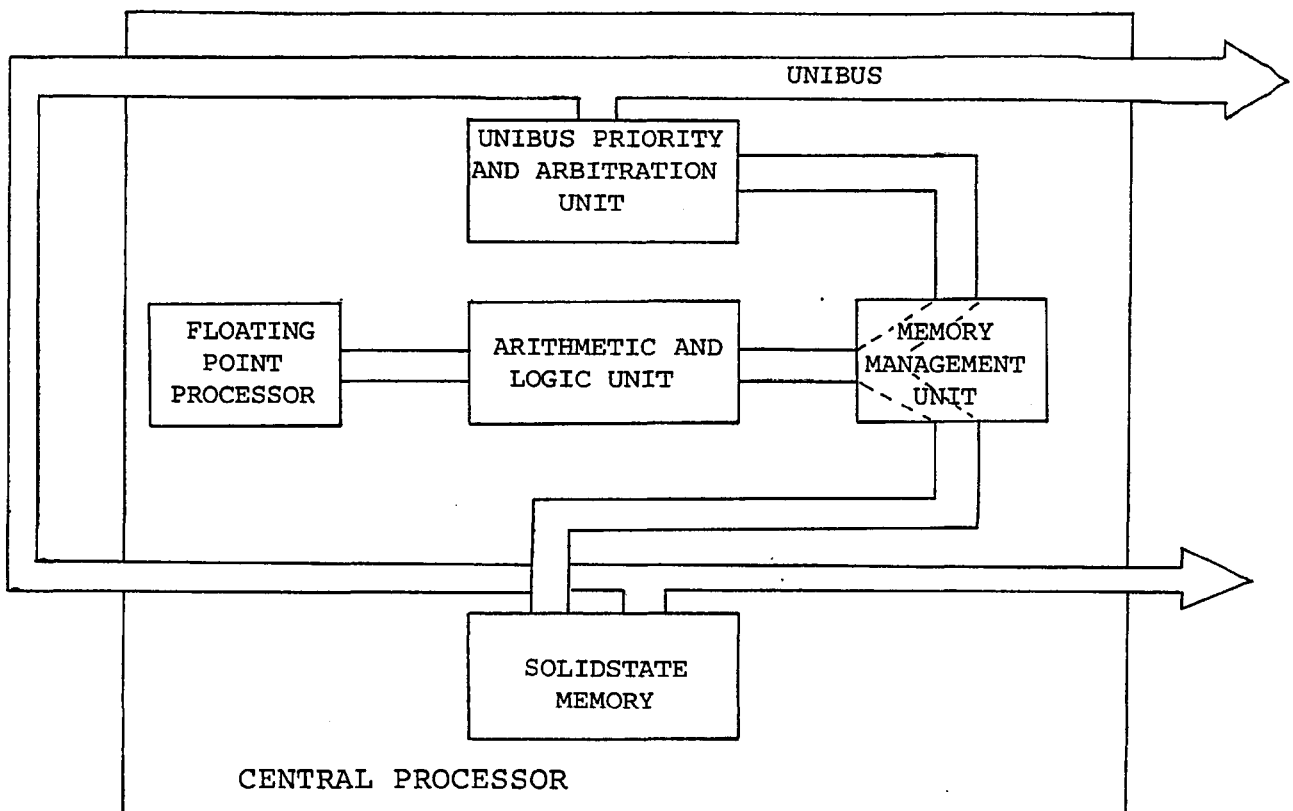


Fig 2.3 PDP 11/45 System Block Diagram

c) The amount of Core Store hardware easily available to the user is usually limited, depending on the size of the computer. The storage can be magnetic cores or semiconductor memories or a combination of both. Further storage is made available in the form of backing or secondary stores, usually magnetic tapes and discs.

Input devices are used to supply the CPU with data and instructions which are stored in the core. The input devices would differ from machine to machine, starting from a simple teletype keyboard for a small machine to a fast card reader in larger installations. Input can also be supplied via backing storage and paper tapes. Output devices accept or record results from the computers operations, and errors if any. The record may be permanent as 'hard copy' printout on a teleprinter or temporary as a 'soft copy' printout on a visual display terminal. Many of the media used for input such as paper tapes, punched cards and magnetic tapes can also be used for 'hard copy' output. All input and output devices are termed peripherals.

2.1.1 Mainframe Based System

A main frame based CAD system will naturally consist of a large computer communicating with man via its various peripherals. Main frame based systems turn out to be expensive to run in fully interactive nature due to the slow reactions of man as compared to that of the machine.

Mini based systems on the other hand, cannot achieve the ultimate goal of CAD yet, since the modern powerful solution techniques require large core store far too great for the mini computers. Thus a compromise is reached by interacting mini and main frame systems at one point where all slow and interactive processes towards the solution of a problem such as mesh generation and other forms of data preparation are performed on a mini based system, and the data transferred via a suitable medium to the main frame system for the solution of the problem. Result obtained from the main computer could be transferred to the mini for analysis. The SAMG system is a minicomputer based system, an extrapolate of the CADMAC-11 system, [20] (Besant et al 1972 Vol. 4 Journal of CAD).

2.2 THE SAMG SYSTEM HARDWARE

The system is built around a Digital Equipment Corporation's (DEC) PDP 11/45 mini computer of 32 K, 16 bit word core capacity, capable of handling multitask operation requiring upto 124 K words of addressable memory space, with the aid of a memory management unit (MMU) - an integral part of the Central Processor. A Floating Point Processor (FPP) is also mounted integrally into the Central Processor, Fig 2.3. All the Central processor units are connected through hardware registers to a high speed bus called the UNIBUS, which provides the communication path for address, data, and control information for all devices on the bus, Fig 2.4.

All devices on the unibus except memory, can dynamically request the UNIBUS to transfer information to another using a scheme based on real and simulated core locations. A detailed description of the architecture of the PDP 11/45 can be obtained from DEC [14] . The computer is interfaced to the peripherals via different control modules.

A DEC Disc Controller is connected to the UNIBUS, interfacing two 1.2 million word moving head disks (RK05), providing fast secondary store. The disk cartridges are removable, so that each user on a single user system is able to remove the disk after usage, hence maintaining security over the user's data and programs. For multiuser purposes a second disk controller for two 2.6 million word discs (RL01) is also connected to the UNIBUS. A third DEC Controller for two magnetic tape drives, one seven track and the other nine track, is also connected to the bus.

The system console is a DEC writer LA-30 and it is controlled by a DEC DL-11 interface. The console is capable of printing thirty characters per second. A high speed paper tape reader and punch capable of reading three hundred characters per second and punching fifty characters per second is also connected to the UNIBUS, using an appropriate DEC Controller. This is very useful for intermachine input/output (I/O) operations. A Data Recording Instruments Component Ltd parallel line printer model 6330 with a maximum speed of

a hundred and fifty characters per second is the most essential listing device on the bus. Two other important interfaces for I/O devices in the system are CAMAC and QUADRASYNC.

[15]

CAMAC was originally developed to provide a common standard interface (CAMAC DATAWAY) into which hardware handling modules can be plugged in. The dataway controller will vary from one computer to another, while the hardware handling modules are independent of the computer. Since the controller and modules are in the form of printed circuit boards plugged into the CAMAC crate, the computer dependency of the controller is no serious handicap. The CAMAC crate normally has space for twenty-four boards, and using the CAMAC interface offers great flexibility for interfacing non-DEC devices, and will permit expansion of the system at a minimal cost. Modules currently attached to the CAMAC crate are: a D-Mac digitizing table, a Tektronics 611 storage tube visual display unit, and the line printer. The SAMG system was originally developed to run with the D-Mac digitizer as the main input unit, but later modified with little effort to use a Talos digitizer which will be discussed later.

[16]

QuadrAsync provides an interface between the PDP-11 UNIBUS and four asynchronous serial communication channels. It has been used in the system for interfacing: Tektronic 4014 terminal, Talos Digitizer, PERKIN ELMER 550 Terminal and a

Flat bed plotter recently developed in this department [18] .

A detailed description of the D-Mac Digitising Table has been covered by Hamlyn [12] .

1. TEKTRONICS 611

The Tektronics 611 storage tube is driven by SEN* vector and character generators mounted in the CAMAC crate, and this provides a visual display unit. There are three levels of intensity, and by correct setting it has been possible to arrange that the lowest level is non-storing. The non-storing mode is used in displaying a tracking cursor which continuously monitors the position of the digitising pen without leaving a stored image of the cross on the screen. The display area is 21 x 16.2 cm, containing 1024 points on every line across the screen in the horizontal direction, and 723 point in the vertical direction. The position of any beam displayed is controlled by start (X,Y) and end (X,Y) coordinates, and a pen code (on or off) for the vector generator to produce the necessary increment or decrement pulse for the display driver to produce a straight line.

2. TALOS 660 DIGITISING TABLE

Unlike the early generation D-Mac digitising table, this is a noise-free solid state digitising table, converting the physical position of the pen or cursor on an active digitiser surface into digital output. With an appropriate

* S.E.N. Electronics Limited

software designed for this unit, the digital output is processed and converted into a format suitable for the Quadra-sync interface. This software has reduced the resolution of the digitiser to ten points per millimetre instead of the manufacturers maximum of forty points per millimetre. There is an absolute origin on the digitiser surface from which all points are mapped. The basic input device is a twelve button cursor with a twelve-foot long cable. Two outputs are produced by the digitiser, one to a visual readout unit, an integral part of the digitiser and the other to the driving software. The visual readout is in octal, and this is very useful since it can be used in obtaining the coordinates of point on a drawing place on the digitiser without a supporting software. The digitiser surface is 127 x 93 cm. In the system, the control program used for bringing application programs into core for execution, is operated by the use of "Menu Commands" from a menu area on the digitising table, hence the reservation of such an area on the digitising surface. The menu area is divided into a number of squares, each identified by a number or a name. Fig. 2.5 shows the digitising table and the menu.

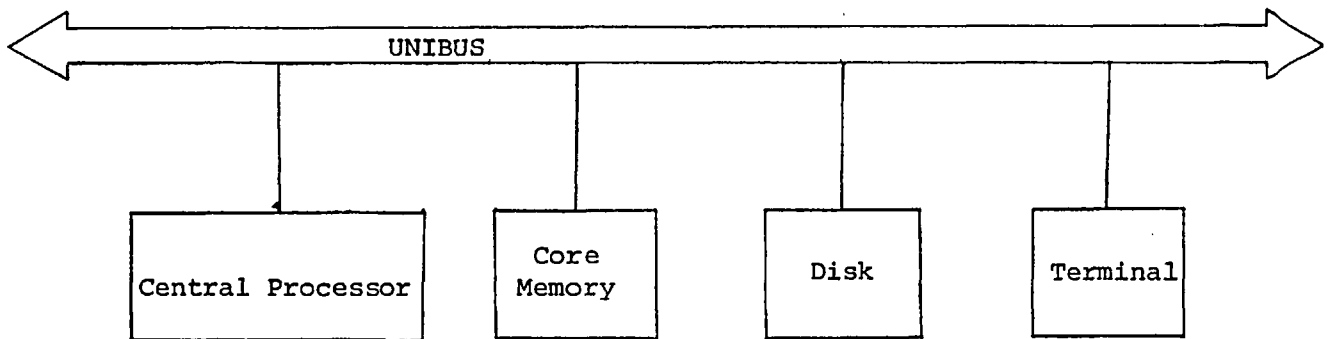


Fig 2.4 The UNIBUS

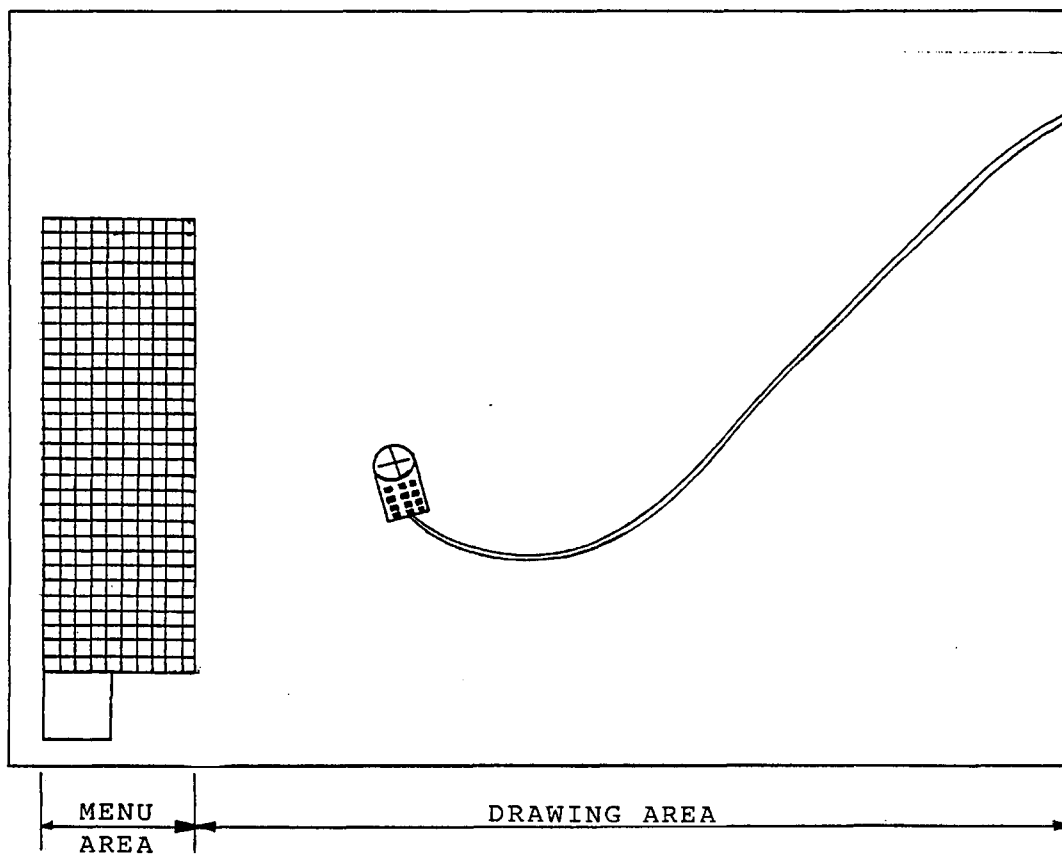


Fig 2.5 Digitising Table

3. PERKIN-ELMER MODEL 550 TERMINAL

This I/O device is a "soft copy" interactive teletype-writer, which communicates with the computer via the QuadrAsync interface in a serial asynchronous mode. Data transmission rates, parity section, stop bit and half and full duplex transmission modes are all switch selectable. The screen has a diagonal size of 25 cm and horizontal size of 20 cm capable of holding or displaying eighty characters per line and twenty four lines on the screen. When the screen is full, there is no automatic erasure, but the last line typed is entered at the bottom line of the screen, and every line being shifted up by one, thus eliminating the first line on top of the full screen before the new entry. Because it is a refresh display terminal, erasures are quite clean and visible.

4. FLAT BED PLOTTER

This is primarily used in the system as a "hard copy" output device which was designed and recently constructed at Imperial College [18] . It is driven by stepper motors via steel belts, and controlled by three microprocessors. It is provided with a First-In-First-Out (FIFO) buffer to store: the pen number, the pen code, and the coordinates of the pen prior to plotting. There are two codes for pen up or pen down, and four pen holders into which different coloured pens can be installed.

Other facilities are: an off/on line plotting mode (where it can virtually be used as a digitiser when on off-

line), an absolute origin and switch for setting paper size from A0 to A4. Paper is held into the surface of the plotter by suction mechanism, acting through tiny holes evenly spaced on the surface.

The plotter has a working area of 120 x 90 cm and a resolution of twenty points per millimetre. The driving software for this plotter as part of the system was designed by the author. The QuadrAsync interface provides the communication channel between the computer UNIBUS and the plotter. The average plotting speed for short vectors is about one inch per second and that of long vectors is about five inches per second, whereas the maximum speed of the plotter is ten inches per second.

5. TEKTRONICS 4014

The Tektronics 4014 computer display terminal is a 'soft copy' interactive tele-typewriter as well as a graphics terminal, which can communicate directly with the computer via an appropriate interface (in this case the QuadrAsync). It can also be connected via Telephone wires and a hook-up consisting of a modulating and demodulating units (MODEM), before being interfaced with the computer ie, a remote control terminal. The terminal can also communicate with various peripheral units and act as a link between these peripherals and the computer - an intelligent terminal. It can be fitted with a separate hard copy unit to produce graphics or characters

on the screen, on a light sensitive paper. It is hoped at the time of writing that this will be the main graphic display terminal when the multiuser graphics package currently being developed is fully operational. A dossier of the structure and operations of this terminal is beyond the scope of this thesis, and can be obtained from the manufacturers.

C H A P T E R 3

CAD SYSTEM SOFTWARE3. INTRODUCTION

In general, the design of a CAD software package tends to depend on the hardware available, although this dependency could be avoided in certain sections of the system software by using standard high level languages that are not machine dependent. This in effect introduces another crucial factor for any CAD system, for if the operating system on the machine does not support the standard high level language, machine dependency cannot be avoided. Another important factor in the development of any CAD system is the specific demand or the scope of the system. The hardware facilities available have been described in the last chapter.

3.1.1 The operating System

An operating system must fulfil the following functions:

1. Organise a collection of hardware into a consistent and coherent whole so that it can be used for development and running of programs.
2. Assist users and their programs to make maximum use of the hardware with minimal programming effort by providing software to supplement and enhance the hardware facilities available.
3. Perform the above functions with the most efficient use of resources especially memory, maximum speed and maximum ease of use.

The operating system used for this work is the standard DEC* Disk Operating system called DOS/BATCH, version 10 MONITOR. The DOS/BATCH Monitor carries out the functions described above in three main areas of activities:

1. Two groups of services to run programs
 - a) Handling of input and output which is the primary function of the monitor, allowing the user to request as much or as little assistance as he needs. Normally the user need not consider the devices the program will actually use when it is executed.
 - b) Providing utility routines for various services. These include loading the program, assisting in data manipulation within the program, and supplying system information.
2. Storing and retrieving sets of data on magnetic media. DOS/BATCH uses a system of named files; a single user can minipulate sets of data easily and conveniently, or several users may store data on one medium without conflict.
3. Memory Management. The Monitor, efficiently handles the competing demands of program, data buffer and Monitor routines for space in memory.

A detailed description of how the Monitor performs these tasks can be obtained from DEC [17].

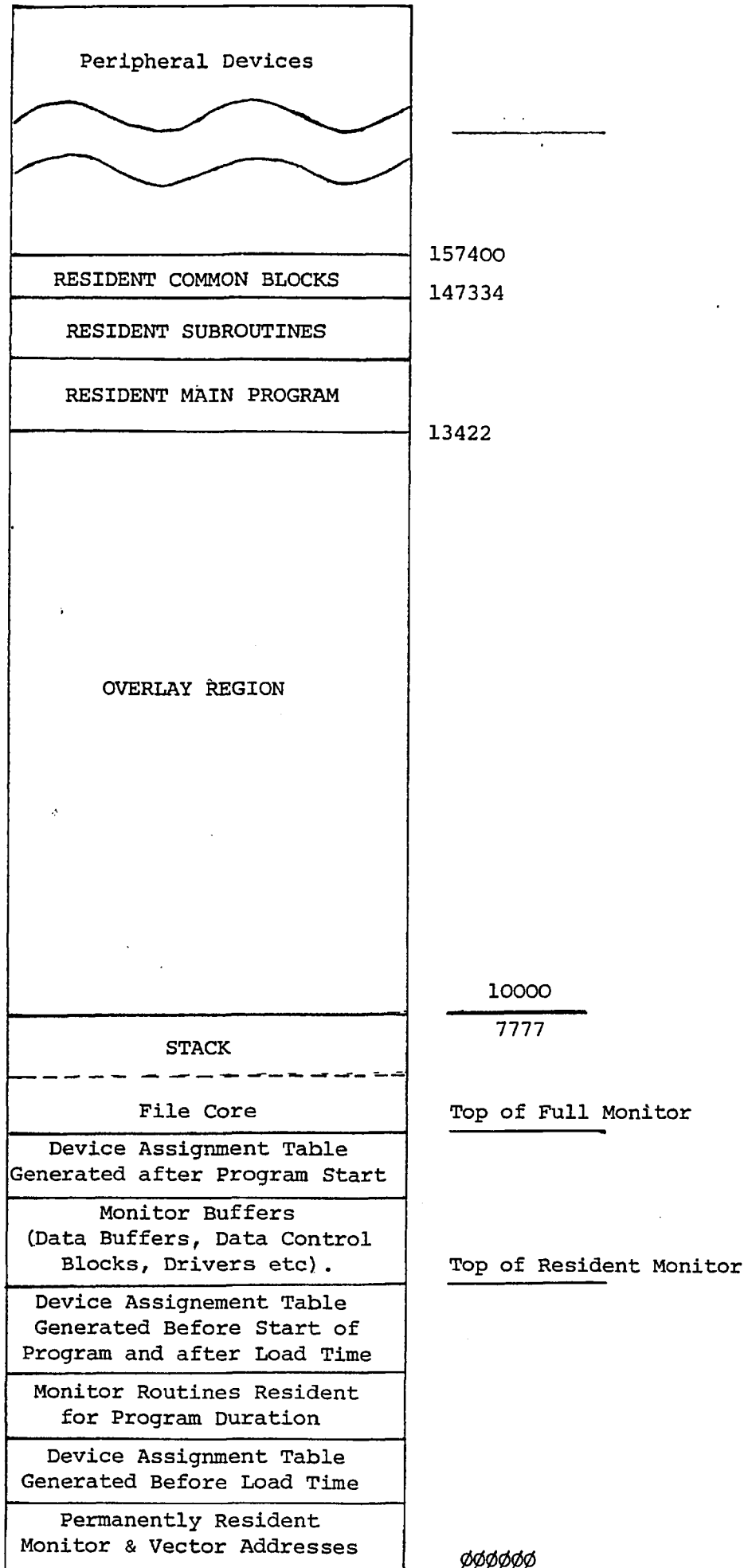
* DEC DIGITAL EQUIPMENT CORPORATION

Out of the 32K words of core store available as described in the previous chapter (section 2.2), the first 4K words of memory are used by the operating system Monitor in providing core space for trap vectors, system communication words, interrupt vectors, buffer space, system input and output handlers, system stack, a small region for resident system subroutines and common blocks, and finally the resident system program. The Monitor also uses the top most 4K words of core store for the status and buffer addresses of all the peripheral devices connected to the computer. The remaining 24K words of core store are available for the development and running of the CAD system, Fig 3.1.

Apart from the standard utility programs provided, the version 10 Monitor supports Macro 11 assembler and Fortran 4 compiler.

3.1.2 Scope of SAMG

The aim for the development of SAMG (a Stress Analysis Mesh Generation package) was to enhance an already existing low-cost CAD draughting system to generate numerical data for Finite Element Analysis programs used in the solution of three-dimensional stress analysis problems. The original graphics package was primarily developed for structural draughting, mainly generating vectors and transferring drawing from the digitising table to disc for storage. From the disc, the drawings can be displayed on the VDU or reproduced on the flat bed plotter.



CORE MAP

FIG. 3,1

General editing and modification of drawings was made very simple by the system. To generate meshes for three-dimensional structures, the new system is required to provide a means of defining these structures in a way that will be easy for the user, and would be clear to the computer. The meshing of these defined structures, computing the coordinates of the nodes created, and creating the coordinate data file and topology data file for all the elements were the main theme of the new system. In addition, the system must be easy to operate, guiding the user via interactive instructions on any of the output peripheral device. The cost of running the system should be small compared to other packages developed for generating three-dimensional mesh data. The system like its predecessor should be open-ended for further development in such a way as to preserve the modularity of the system by making new modules compatible with existing ones. The most important new modules added to the old graphics package, and the enhancement routines, all resulting into SAMG will be seen mainly in chapters 4 and 5. In due course the mesh generation modules will be made available on a multiuser graphics system currently being developed in this department. It is hoped that SAMG on further development could be used also as a post-processor for analysing the results of Finite Element solutions.

3.2 THE GRAPHICS SYSTEM

Due to ^{the} small core store available for the graphics system, an overlaying technique is used in running the system.

This technique involves the dividing of the available core store into a resident and an overlay areas, Fig 3.1. The resident area contains a small executive, some commonly used subroutines, named common blocks used for storing intermediate results, and buffers for three disc resident files called work spaces. This area of core store is made as small as possible, allowing the maximum amount of core space for the overlays. The fast overlaying technique implemented in the system is different from the one provided by DOS/BATCH, and it was developed at Imperial College by Hamlyn [12] . The load module of each program written as an overlay is assigned a number in addition to its name, and written into a library of a contiguous random access file SMGLIB.OVL on the system disc by a stand-alone overlay building program OVAL. The first two blocks of the library contains a directory of all the overlays, including a name, number, start record numbers, and length for each overlay. Making it possible for any overlay in the library to be called by software and some by hardware. By software, any overlay can call another into core, overlaying itself, but if the results from the overlay called, stored in the resident common blocks, are important to the calling overlay, we would want it back in core, and to do this with speed advantage, a FILO (First-in-last-out) stack like the PDP-11 stack is created in the general software for stacking overlays, Fig 3.2. This provides two ways of calling an overlay into core by software, either by a direct called using the overlay loading program OVLINK or by first putting the overlay in the stack and calling a return from the current overlay

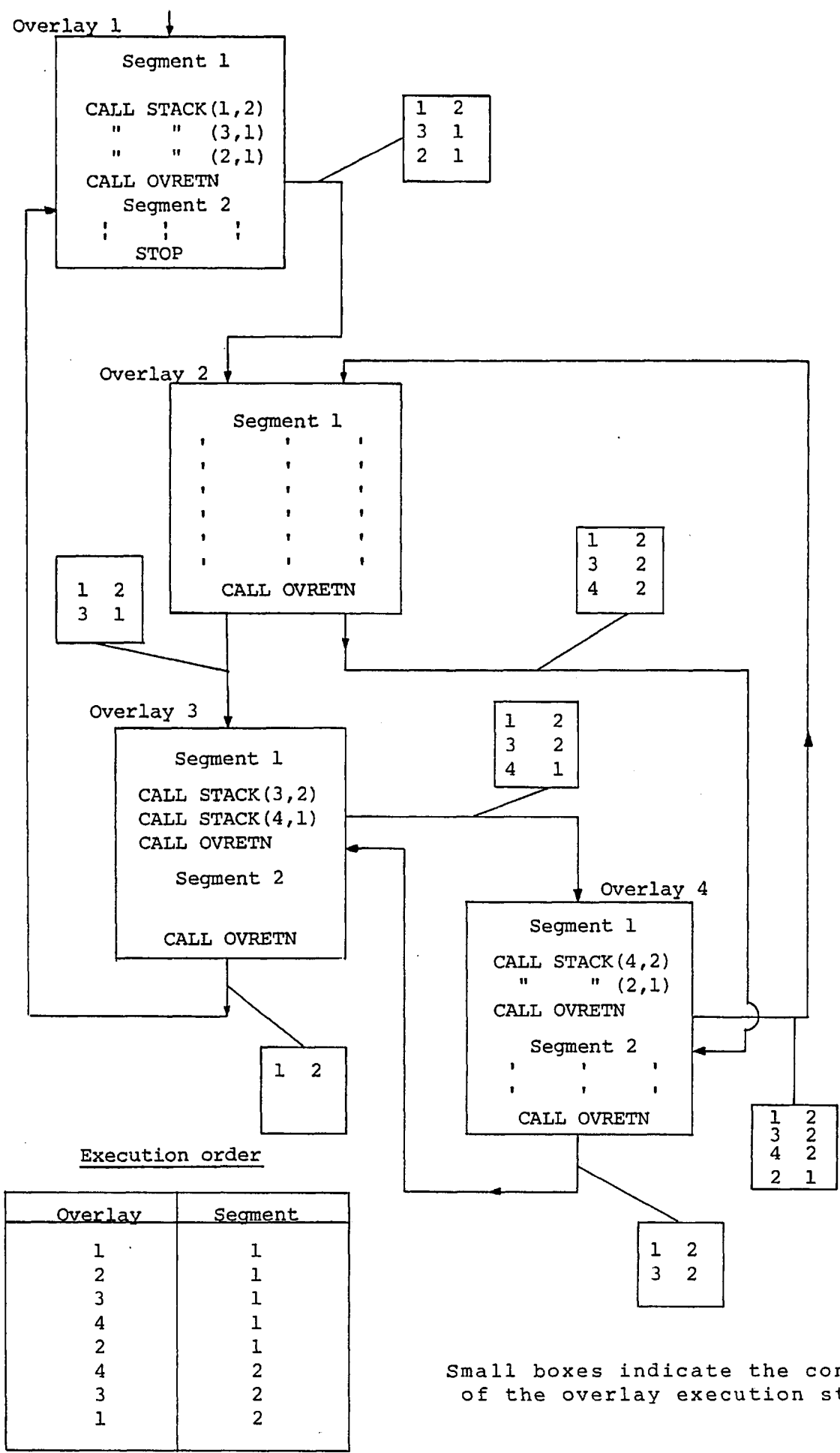


Fig 3.2 Example of overlay stacking

in core. The hardware call is the means by which the user controls the system and calls overlays into core from the digitising table by use of "menu" commands.

3.2.1 The Menu

The active area of the digitising table is divided in to two: a drawing area and the menu area. The "menu" is a 10 x 30 matrix of 20 mm square occupying the left most part of the active area and ninety millimetres above the lowest horizontal line active. Whenever any of the twelve buttons of the digitiser pen is pressed during operation of the system, either of two things will happen: the point is outside the active area and a light on the digitiser panel will indicate it, or the point will be accepted by the system executive. If the coordinates of the point accepted is within the menu area, the system executive recognises this as a menu command and calls a subroutine MENAN to decipher which of the three hundred squares above which the pen was pressed. By sending this number to another subroutine (which in fact is a software menu) MENMAP, the number of the overlay corresponding to the square is obtained and the overlay is loaded into core for execution. In this way the user can issue "menu commands" without turning to the system terminal continuously.

Should the pen be pressed over the drawing area of the table, the coordinates of the point are accepted and used as required, depending on the number of button press. The functions of the button number will be seen when we consider the system modules.

3.2.2. System Modules

The graphics system is made up of a set of modules and utilities, some of which are overlays and others subroutines which are transparent to the user. In fact some of the overlays are also transparent to the user, like the SETUP module which on first entry of the system is called into core by the resident executive.

1. SETUP is used by the system executive to initialize the system parameters, any of which can be reset later by menu command. These parameters are: the table origin, the drawing origin, the input and output scales, the skew defining the angle between drawing and table axes, the grid factor for defined step increase, and the alphanumeric size. It also initializes the input and output pointers on all three random access working files (see DATA STORAGE AND FILE HANDLING). Thereafter control is given to DIGOV the User Operating Monitor UOM, by SETUP overlaying itself with the UOM.

2. DIGOV is the digitising module or overlay and the heart of the system containing the following: Background loop, drive mode, interrupt handler, menu command handler and symbol handler. The background loop is responsible for reading and converting the table coordinates to drawing coordinates, adjusting these coordinates if there is a grid factor and/or skew factor, detecting the number of the button pressed on the digitising pen, displaying the coordinates of pen relative to last point digitised, and a tracking cursor representing the current position of the pen on the screen. In drive mode,

the cursor is driven from the last point digitised in discrete intervals in cartesian coordinates, and this enables accurate insertion of exact dimensions. The movement and position of the cursor and the value of the drawing coordinates are controlled from a 90 mm square at the bottom left hand corner of the digitising table.

When the operator presses any of the buttons of the digitising pen and it is detected by the background loop, control is passed to the interrupt handler. The effect of pressing button 1 is either to load on overlay into core if the button is pressed in the menu area, or to get the coordinates of a point in the drawing area. Button 2 breaks the continuous plotting of line from the previous digitised point to the current point. Button 4 transfers control to the "WINDOW" overlay while button 8 is used to switch in and out of drive mode. Button 7 is the Find Mode with which a previous point can be accurately digitised when the pen is pressed near the required point.

As said earlier if button 1 is pressed in the menu area, control is passed to the menu handler which identifies the particular overlay number that should be called if the system is in line mode. On the other hand, if the system is not in line mode a CALL OVRETN is executed and the next stacked overlay will be called. The command area of the menu is arbitrarily divided into two sections: System Command area and user Command area; and in order that the position of

particular command squares may be rearranged at will without having to change many programs, a mapping system is used. This is shown schematically in Fig 3.3.

Symbols are used to describe graphical data items of easy geometric definition such as circles, rectangles etc., which are frequently used in the system. Special routines have been written to generate them with a minimum number of input points; to describe dimension specification, and position within the workspace.

3.2.3 Data Storage and File Handling

The term 'data storage' is defined as the process by which information is placed on a medium in machine-retrievable form for subsequent use on the same or another system. The first requirement of any storage medium is reliability. Other desirable characteristics are:

1. High speed of access and transfer
2. High capacity
3. Portability
4. Low cost

The DOS/BATCH system supports four magnetic media: disc (both fixed and interchangeable), DECtape, standard magnetic tapes and cassettes. Paper tape input and output devices may also be used for data storage, [17]. The relative strengths of each medium are shown in the following table, in which the devices are ranked from 1 (fastest access time) to

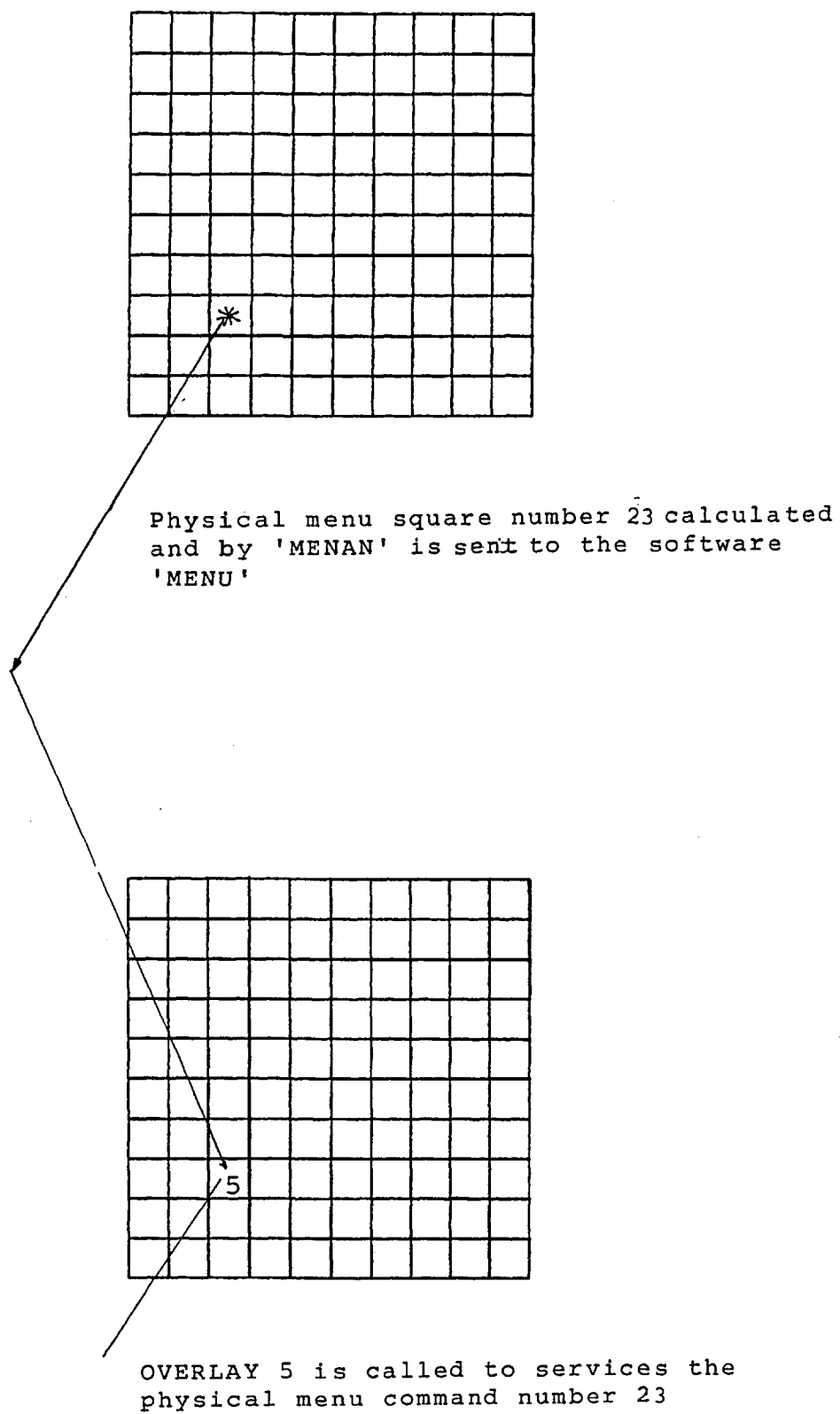


Fig 3.3 Menu Mapping "MENMAP"

5 in respect of each characteristic.

TABLE 3.1

	DISC	DECtape	Magtape	Cassette	Papertape
Access speed	1	2	3	4	5
Transfer rate	1	2	3	4	5
Capacity	2	3	1	4	5

For reasons of speed the only type of file the system uses directly are three random access files on a contiguous region on the moving head disc unit, as working files, and other random access files as mass storage files. A total of sixteen files may be defined, and for two disc units the files could be arbitrarily spread between them. In the system, the three working files SMGDAT.RA1, SMGDAT.RA2 and SMGDAT.RA3 are created on disc unit zero the system disk, and these files correspond to three named resident common block which are used as their buffers. At the start of the system, these random access files and their resident common buffers are initialised by the SETUP module which looks up all the files to check their validity and length, and sets a record pointer (NREC), and data point (IDP) for each file to zero. For each point digitised as a coordinate, the data is written and stored in the appropriate resident buffer by a statement:

```
CALL SETR (IFN, ICODE, f(COOR))
```

and read by a statement:

```
CALL GETR (IFN, ICODE, f(COOR))
```

where:

IFN - is the random access file number
 ICODE - is the pen code of function code (Table 3.2)
 f(COOR) - is the coordinates of the point
 X, Y for two-dimensional work space, and
 X, Y, Z for three dimensional work space.

For each call to either of the above subroutines, there is an automatic increment of IDP the data pointer by one. All the workspace buffers are each one block or 256 words in size, thus there can only be thirty-six three-dimensional data and four words to spare since each data consists of seven words one for the code and two each for the three coordinates. For 2D data there can only be fifty-one and one word to spare per full buffer. When any of these buffers is full or cannot possibly accommodate any incoming data in the available space, the content of the buffer is written to the corresponding working file on the disc by a statement:

```
CALL RAWRIT (IFN, IREC, ARR, NW)
```

and read by

```
CALL RAREAD: (IFN, IREC, ARR, NW)
```

where:

IFN - as before, the random access file number
 IREC - is the record number to be written or read
 ARR - is the array in core from which data is to
 to be read or written
 NW - is the number of words for transfer.

Since there are 256 words in each buffer, if the number of variables for transfer is not specified then 256 words (128 single precision real or integer variables) will be transferred by default. Again for each call to the above subroutines, the record pointer (IREC) is incremented by one.

The file handler allows data transfer between work files and mass storage files. SAMG uses only one random access file for mass storage and two working files, one for 2D and the other for 3D. The third working file, which can be access directly, is mainly used by the system for storing ^{the} perspective view of the 3D working file. SMGDAT.MS1 is used to hold a set of 128 storage files. The five main operations carried out by the file handler are: transfer of data between the buffers and the working files, addition of data to a working file, transfer of data between working and storage files, display of files and deletion of files. Any of the working files could be stored on another disc or magnetic tape for future display or modification.

lCODE	FUNCTION	f(COOR)
0	END OF DATA	NOT SPECIFIED IN CALL
1	PEN UP	COORDINATE X,Y and/or Z
2	PEN DOWN FROM PREVIOUS POINT	,,
3	START MACRO OR SYMBOL	MACRO NUMBER
4	END MACRO OR SYMBOL	,,
:		
15	NULL DATA	NOT SPECIFIED IN CALL

TABLE 3.2

For crash recovery a separate overlay RECOVER has been built to recover the whole data on the working files in the event of a system breakdown. This recovery overlay is distinct from that used in recovering a previously stored working file or picture.

3.2.4 Displaying, Plotting and Windowing the Workspace.

The modules for displaying on the Tektronics 4014 and plotting on the new flatbed plotter were developed by the author to replace the previous modules written to drive the Tektronics 611 VDU and a DC plotter via the CAMAC crate. These new modules are written to drive the units employing a QuadrAsync interface mentioned in Chapter 2. These modules are responsible for displaying and plotting of data in the working files, and for display purposes in the 3D work, data is arranged in three separate random access high speed transfer files. These are

known as FILE 1, FILE 2 and FILE 3. FILE 1 contains the true 3D data describing the object, and this data can be viewed as a wire frame structure by a suitable projection produced by a JOYSTICK program (which will be discussed later) and stored in FILE 2. The data structure of FILE 1 and FILE 2 are identical in sequence but the latter has its X and Y coordinates in terms of the screen value and the Z coordinate in terms of the actual spacial depth produced by the perspective transformation. FILE 3 is the 2D file containing the coordinates of the actual points digitised, in other words the actual blue print of the digitised artwork.

The display of each file is controlled by a value specified in the workspace display or plot routine, designated a value ND. The program DISALL display on the VDU while PLTALL plots on the flatbed plotter and in both cases:

```

ND = 1    converts and produces file 1
ND = 2    displays or plots FILE 2
ND = 3    displays or plots FILE 3

```

While displaying or plotting, the presence of any symbol or macro in the data calls in the appropriate symbol overlay to generate the symbol for display or plotting.

The scale of the display and plot is determined by parameters set in the windowing module. Setting a left-hand window and a right hand window will enlarge the rectangular

area described by these points to fill the whole screen. In the event of a situation where there are vectors outside the window area the operation of producing the drawing within the window area is sped up by a clipping process. A negative window or a decrease in the size of a drawing for displaying or plotting purposes is achieved by using the SETUP module to decrease the output scale or increase the input scale, both of these operations giving the same effect.

3.3 APPLICATION PROGRAMS

The last section described forms the foundation of a general purpose graphics package on which a host of application programs may be built. An application program is in general a module or a group of modules designed to perform a specific task, for instance; draughting, animation, printed circuit board and pipe routing, mesh generation and a lot more. The development of each application program is tailored by the specific needs of the application and the compatibility of the module developed with the existing modules of the graphics package. The application programmer only needs to know the basic data storage and file handling routines, and the program communication arrangements. When application programs are built into overlays, the application programmer must remember that any time an overlay is called and loaded into core, the previous contents of the overlay area are destroyed. In order to communicate between overlays, three methods are employed:

1. The storing of information in the resident common

area.

2. The storing of information on disc file which is later accessed by another overlay.

3. The storing of information on the overlay execution stack.

In all or most application programs all the modules described in section 3.2 will be found necessary while in certain applications there will be extra basic modules added to the nucleus of the graphics package. A brief summary of two of the existing packages that have sprung from the basic system will be discussed.

3.3.1 The use of the system for draughting

The inputs to the system for draughting take place in one of a number of symbols, generally selected by menu command except in the case of line mode which although can be set by menu command is the default setting on first entry to the system. The symbols provided include: Line mode, Arc mode, Circle mode, Alphnumeric mode, Dimensioning mode and Rectangle mode. With these modes any figure can be digitised and labelled. There are utility routines to edit points, lines and whole sections of drawing (symbols and macros).

Two extra modules for handling macros and levels are provided and with their usage, drawings can be split into different levels and later joined, or parts of drawings consisting of several symbols arranged in any order could be

stored as macros. These macros could be scaled, rotated and translated to any position in the working space to form parts of a drawing being developed as shown in Fig 3.4a and 3.4b.

Miscellaneous facilities for continuous digitising, debugging of the working files, curve fitting and perspective and isometric viewing of the workspace have been developed in the system for draughting. A detailed description of the use of the system for draughting can be obtained from Hamlyn [12].

3.3.2 The use of the system for animation

A two dimensional general animation graphics system was developed, based on the original graphics package by Yi [19]. Although some of the original graphics facilities for data manipulation were incorporated, many of the input and output facilities were found inadequate for handling large amounts of pictorial information needed in the case of animation. A number of special graphics input facilities were added to facilitate the construction of pictures. The output stages in turn were made more efficient by using faster algorithms to generate at a high rate and accept picture input parameters in animation terms.

This particular application software included a suite of special functions used for the manipulation of pictures using a hierarchical structure of data to facilitate the manipulation of pictures expressed in terms of pictorial primitives, (Fig 3.5).

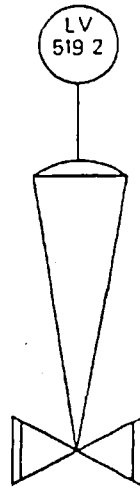


Fig 3.4a MACRO

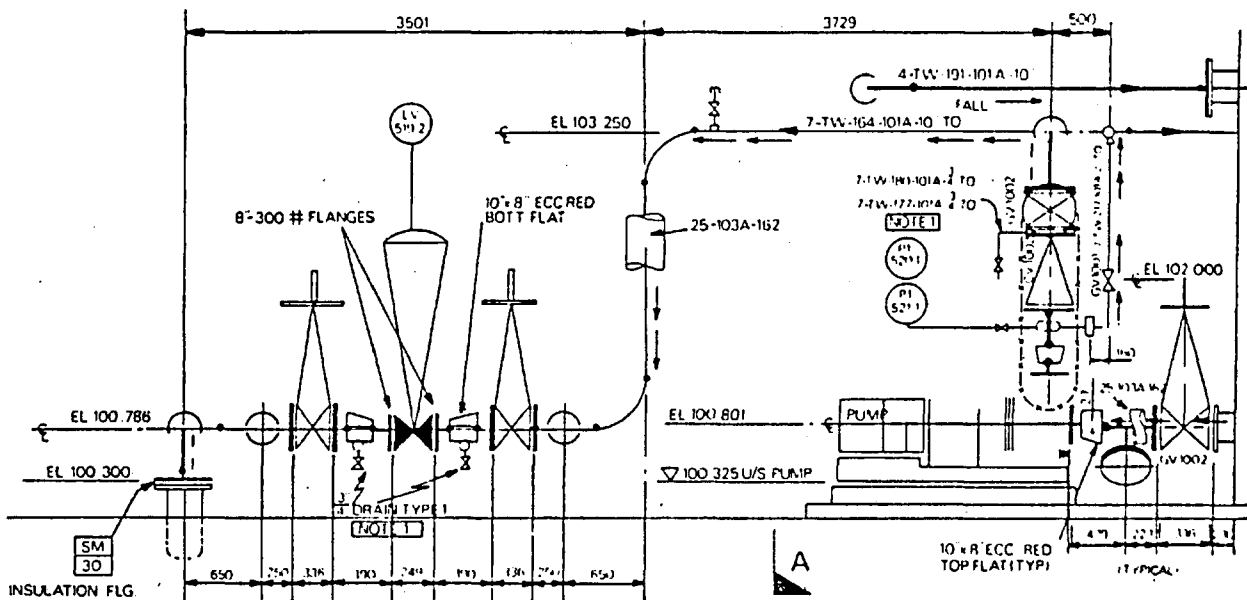
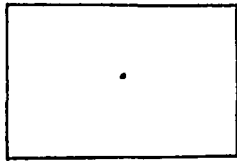
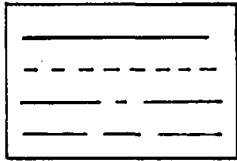


Fig 3.4b Drawing, including the Macro



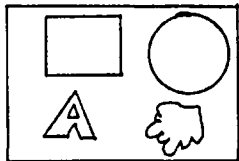
POINT

(the basic element)



LINE

(4 types)



SYMBOL

(standard or user defined)



MACRO

(in one file - can be made up of symbols)



COMPONENT

(one or more files - each corresponding to a different cel level)



FRAME

(more than one file made up of components and background)

Fig 3.5 Picture hierarchy

These pictorial primitives were used singly or combined to create pictures of increasing complexity. The animation package was successfully used in producing films and a detailed description of the package and its capabilities can be obtained from Yi [19] .

Finally most of the software that will be described in the next two chapters are application programs developed or modified by the author for the generation of meshes, computing the coordinates of the nodes and preparing the topology data for finite element analyses. The programs were all developed and incorporated into the basic graphics package to form a Stress Analysis Mesh Generation package (SAMG) .

C H A P T E R 4

AUTOMATIC MESH GENERATION FOR 2D AND AXISYMMETRIC SHAPES

For the finite element analysis of simple plane strain structural problems, most users are involved in the writing of special programs to generate the mesh for each particular problem. When the shape of the model is complicated, it is nearly impossible to develop an efficient program to generate the mesh required for the analysis. Generally, the generation of meshes within simple geometric shapes such as quadrilaterals, triangles and circles is trivial, thus the basic building blocks of any complex geometric shape could be taken as any of the three simple shapes or a combination thereof. It can then be seen easily that, the use of CAD techniques in the generation of meshes for complicated shapes depends on the ability of the user, to successfully divide the shape into series of quadrilaterals and part circles within the boundaries of the structure. Once this is accomplished, the input of all the necessary parameters of any of the quadrilaterals for generation of meshes and the joining of all the quadrilaterals and part circles into producing the initial shape, will be carried out interactively by man and machine. For any quadrilateral, once it has been defined by digitising the four corners in a clockwise or anticlockwise direction; the enclosed area may be divided into any number of smaller quadrangles or triangles, depending on the number of elements required. Fig 4.1 shows a quadrilateral divided into sixteen quadrilateral elements.

4.1 Theory

A quadrilateral with four sides of different lengths as shown in Fig 4.2e is generated through successive modifications, starting from a unit square, Fig 4.2a. If the global coordinates of a typical node shown in Fig 4.3a are X_i and Y_i , then the new values of these coordinates in the quadrilateral ABCD in Fig 4.3b, after the first modification x_i and y_i are obtained from the following equations:-

$$x_i = X_i * L \quad (4.1)$$

$$y_i = (d - x_i * \tan\beta - x_i * \tan\alpha) * Y_i + x_i * \tan\alpha \quad (4.2)$$

where d and L are the depth and length respectively of the quadrilateral ABCD, and α and β are the angles of the lines AD and BC respectively, to the horizontal as shown in Fig 4.3b. These angles are computed internally by the mesh generating program using the four corners of the quadrilateral supplied by the user.

Consider a general case where n_x is the number of nodal points per lengthwise row and n_y the number of nodal points depthwise column, the total number of nodes will be $n_x * n_y$. The total number of triangular elements will be $2(n_x - 1)(n_y - 1)$ and that of quadrilateral elements will be $(n_x - 1)(n_y - 1)$. Let i_x and i_y be used in counting the nodes in the increasing directions of the length and depth respectively where

$$1 \leq i_x \leq n_x$$

$$1 \leq i_y \leq n_y$$

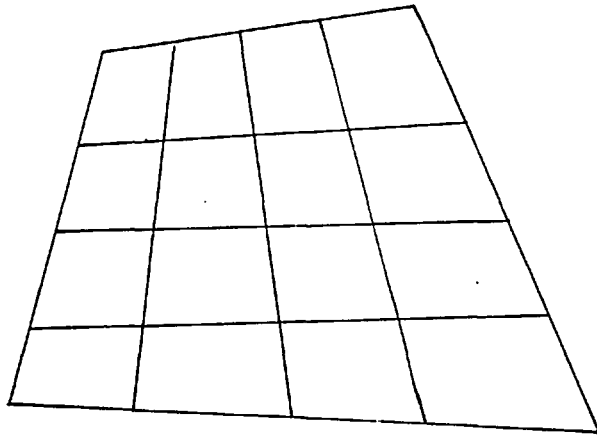


Fig 4.1 Simple Mesh in a quadrilateral

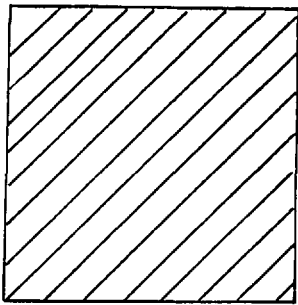


Fig 4.2a

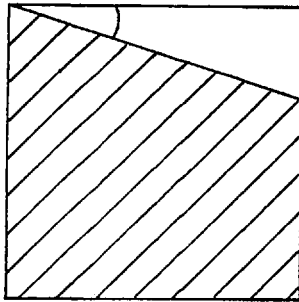


Fig 4.2b

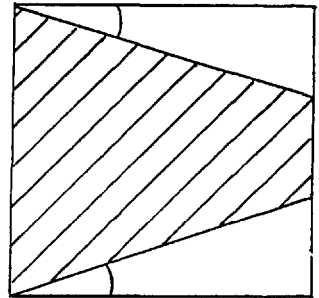


Fig 4.2c

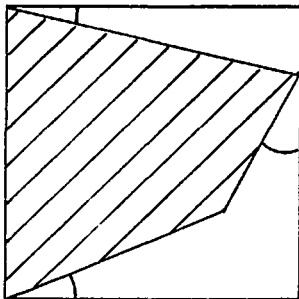


Fig 4.2d

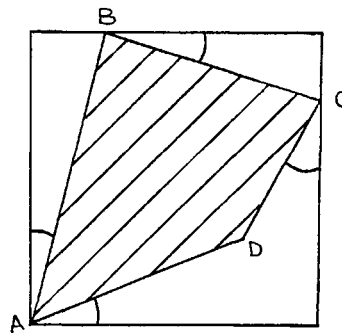


Fig 4.2e

Steps of Successive Modification

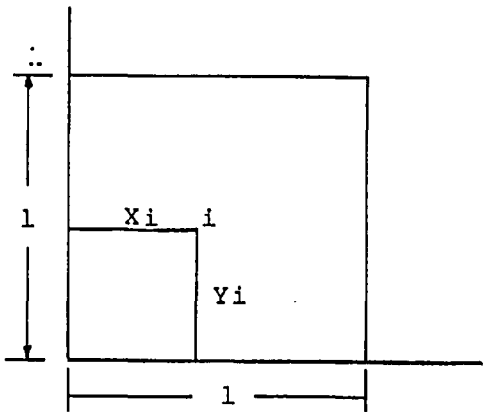


Fig 4.3a
Global Coordinates

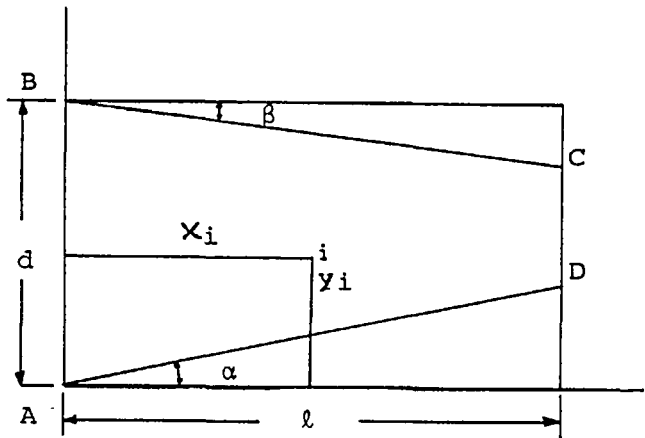


Fig 4.3b
Modified Coordinates

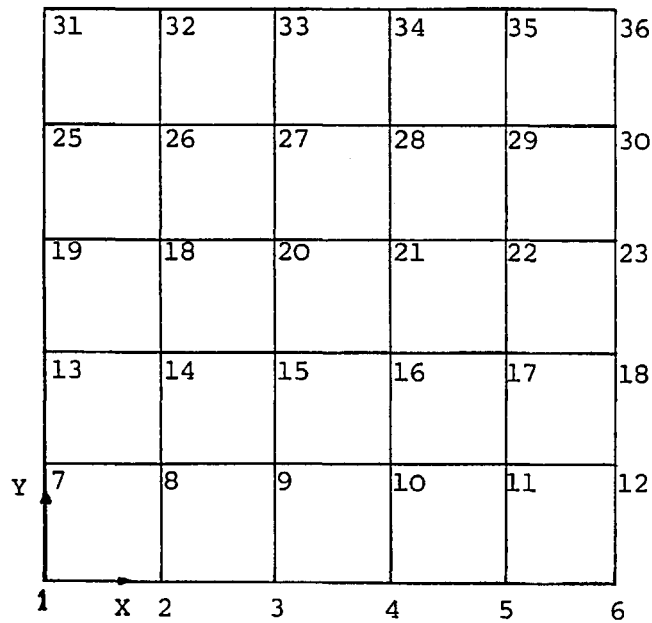


Fig 4.4 Uniform Mesh

If the numbering of the nodal points is such that it starts from $Q = 0$ and goes along the length before starting on a higher row (ie. increasing in depth) then the number of a typical nodal point i can be derived from the corresponding values of the counters [21] .

$$i = (iy-1) \cdot nx + ix$$

Assuming that the mesh has unit overall dimension on all sides and the origin of the global coordinates is at the bottom left hand corner of the quadrilateral, then the global coordinates of a typical node in the uniform mesh shown in Fig 4.4 will be [21] .

$$Xi = \frac{ix - 1}{nx - 1} \quad \text{and} \quad Yi = \frac{iy - 1}{ny - 1} \quad (4.3)$$

The next modification towards getting the actual shape of quadrilateral will be, to account for the angle of inclination of the line CD in Fig 4.2d. To find the new position of each node on line CD, consider Fig 4.5 where CD is replaced by line 3-4. It is necessary to determine the horizontal distance between each new node and the previous corresponding node. Let BB be the horizontal distance between any nodal point on line 3-4 and its previous position on line 3-5. The value of BB for node P can be obtained by solving the triangle ABC in Fig 4.6 as follows

$$m = Y_3 - Y_C$$

where y_c can in general be the y value of each node when it is generated inside the quadrilateral 1235 Fig 4.5.

$$\begin{aligned}
 DB &= m \cos \gamma \\
 DC &= m \sin \gamma \\
 DA &= DC \tan (\alpha + \gamma) \\
 AB &= DA + DB \\
 &= m \{ \cos \gamma + \sin \gamma + \tan (\alpha + \gamma) \} \\
 BB &= AB \sin \gamma \\
 &= (y_3 - y_c) \sin \gamma \{ \cos \gamma + \sin \gamma + \tan (\alpha + \gamma) \}
 \end{aligned}$$

Each nodal point on line 3-4 is at a horizontal distance E from line 1-2 and E can be derived from the following equation

$$E = K - BB$$

where K is the horizontal distance between the first and third corners of the quadrilateral.

To obtain the X - coordinate of each node x_i , E for the node will be multiplied by a factor representing the global coordinates of the node.

$$x_i = E \frac{x - 1}{n_x - 1} \quad (4.4)$$

The Y - coordinate of each nodal point y_i will be obtained from

$$y_i = Y'_i - \delta \tan \alpha \quad (4.5)$$

where $\delta = X'_i - x_i$

X'_i = the X - coordinate of the node when the modification stage has reached Fig 4.5 quadrilateral 1235

Y'_i = the Y - coordinate of the node in Fig 4.5

For example when $i = 8$ X_8^1 and Y_8^1 are the coordinates of node number eight in Fig 4.7a. The new coordinates of node number eight will be x_8 and y_8 after applying the above modification. The resulting position is shown in Fig 4.7b.

The final modification will be the application of a shift to line 1-5 in Fig 4.8a while the coordinates of the nodal points on lines 1-4 and 3-4 remain unchanged. Graphically, by drawing lines parallel to the X axis from all the nodal points on line 1-5 and 5-3, the intersection of these lines with line 1-2 and 2-3 respectively will give the new nodal positions for the quadrilateral 1234 as shown in Fig 4.8b. Effectively there will only be a shift or modification on the X coordinates of all the nodal points involved while all the Y coordinates remain unchanged. The new value of the X - coordinate for each nodal point will be obtained as follows

$$X_i = SD + DI \frac{x_i - 1}{n_x - 1} \quad (4.6)$$

where $SD = y \tan \theta$

$DI = x - SD$

$x =$ the previous value of x_i in Fig 4.7b

$y = y_i$, the value of the Y coordinate

The theory so far discussed was extensively used by Ghassemi [10] in the development of MESHGE - a two dimensional mesh generation program. Following similar modifications to arrive at a suitable technique for three dimensional meshing

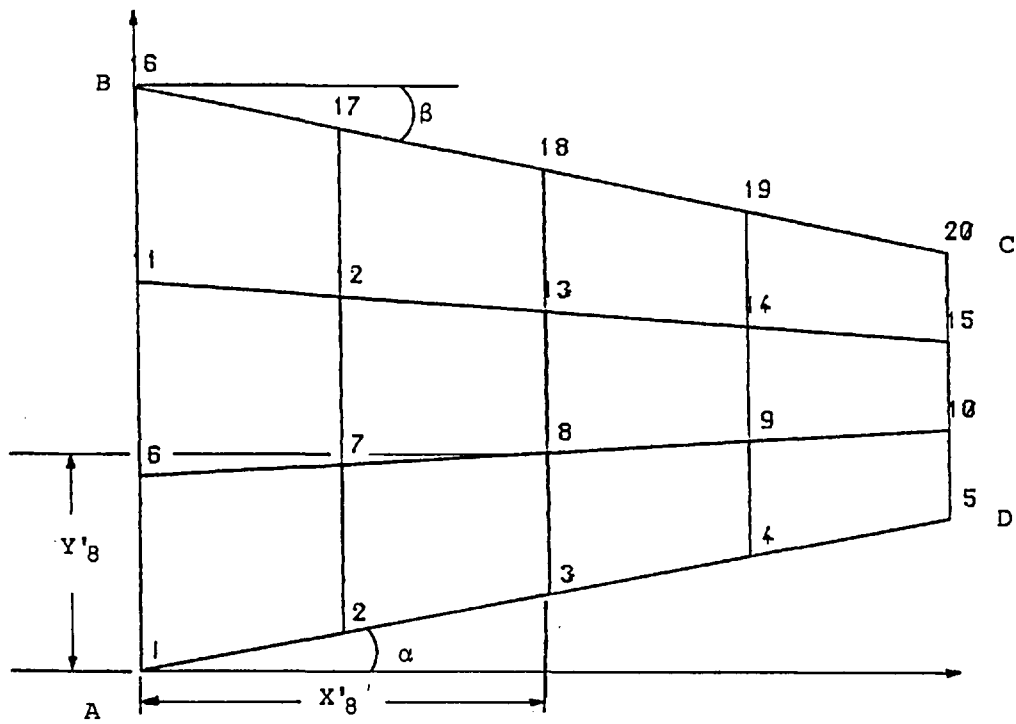


Fig 4.7a Mesh Modification

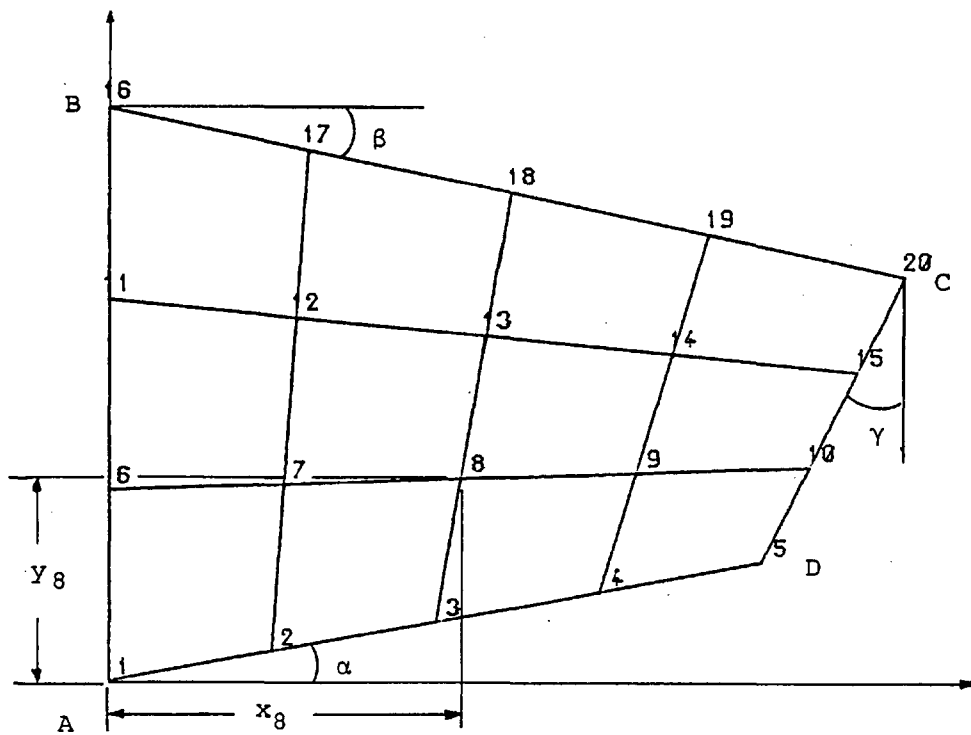


Fig 4.7b More modification

was impractical due to the sheer size of the program, since MESHGE alone occupied 55 blocks or 13.75 kilo words of memory at run time. It became necessary to develop or formulate a simpler algorithm to generate the two dimensional mesh for two reasons:

- a. To make the program as small as possible, since the space occupied at run time is the same to that in the overlay library. By so doing there will be enough room in the overlay library for new overlays to be inserted - those for three dimensional meshing.
- b. The development of a similar program for three dimensional meshing can easily be derived from that of the two dimensional meshing program.

4.1.1 Coordinate Geometric Algorithm

Consider the quadrilateral in Fig 4.9, all we need to know are the coordinates of the four corners of the quadrilateral and the number of nodal points along line AD and AB. Four lengths are then defined:

$$XB = CX(D) - CX(A)$$

$$XT = CX(C) - CX(B)$$

$$YL = CY(B) - CY(A)$$

$$YR = CY(C) - CY(D)$$

where $CX(N)$ is the value of the X - coordinate at corner N
and $CY(N)$ is the value of the Y - coordinate at corner N

Like the previous algorithm the numbering of the nodal points is from left to right starting from the bottom left hand corner and progressing to the top. Thus the typical node number is obtained from the equation

$$I = (IY - 1) NX + IX$$

where $1 \leq IX \leq NX$

$$1 \leq IY \leq NY$$

IX and IY are the counters as before

NX and NY are the total number of nodes along AD and AB respectively.

In our example I goes from 1 to 4 along line AD . It can easily be shown that when I is 3

$$X(3) = CX(A) + \frac{2}{3} XB \quad (4.7)$$

where $X(3)$ is the value of the X coordinate at node number three. Rewriting equation (4.7) in a general form for every nodal point on line AD , we obtain the following equation

$$X(I) = CX(A) + \frac{I - 1}{NX - 1} * XB \quad (4.8)$$

and

$$Y(I) = CY(A) + \frac{I - 1}{NX - 1} * \{CY(D) - CY(A)\} \quad (4.9)$$

where $X(I)$ and $Y(I)$ are the X and Y coordinates of the node I on the line AD , I going from 1 to 4.

Since the ratios $(I-1)/(NX-1)$ and $(I-1)/(NY-1)$ appear in the computation of every single node for which these ratios have different values, they are defined as follows

$$XIR = \frac{I - 1}{NXL} \quad (4.10)$$

$$YIR = \frac{I - 1}{NYL} \quad (4.11)$$

where $NXL = NX - 1$

and $NYL = NY - 1$

From equation (4.8) it can easily be seen that for any nodal point along the line AD, all we need to know in order to find the coordinates of the point are: the coordinates of the point A, those of the point D, and the ratio of the node's position along the line AD.

Considering any nodal point within the quadrilateral ABCD, for example the node numbered 10, its X - coordinate will be one third of the horizontal length between node numbers 9 and 12, added to the X - coordinate of node number 9. The Y - coordinate will be two thirds of the vertical length from node 3 to node 15, added to the Y - coordinate of node number 3.

For node 10

$$IX = 2$$

$$IY = 3$$

$$\begin{aligned} I &= (IY - 1) * NX + IX \\ &= \quad \quad \quad 8 \quad + \quad 2 \end{aligned}$$

$$XIR = (2 - 1)/3 = 1/3$$

$$YIR = (3 - 1)/3 = 2/3$$

$$X(10) = X(9) + \frac{1}{3} [X(12) - X(9)] \quad (4.12)$$

$$\text{also } X(10) = X(2) + \frac{2}{3} [X(14) - X(2)] \quad (4.13)$$

To rewrite equation 4.12 or 4.13 in a more general form for any nodal point within the quadrilateral we need to find a relation between the three nodes involved, the ratios defined, and the total number of nodes in the lengthwise and depthwise rows. Hence considering the equation 4.13 we obtain the following relationship:

$$\begin{aligned} \text{For } X(10) , \quad I &= 10 \\ IX &= 2 \\ (NY - 1) * NXL + IX &= IP = 14 \end{aligned}$$

By using the above relationship equation 4.13 can be rewritten as

$$X(I) = X(IX) + YIR [X(IP) - X(IX)] \quad (4.14)$$

But from equation 4.8,

$$X(IX) = CX(A) + \frac{IX - 1}{NX - 1} * XB \quad (4.15)$$

From equation 4.10

$$\frac{IX - 1}{NX - 1} = XIR \quad (4.16)$$

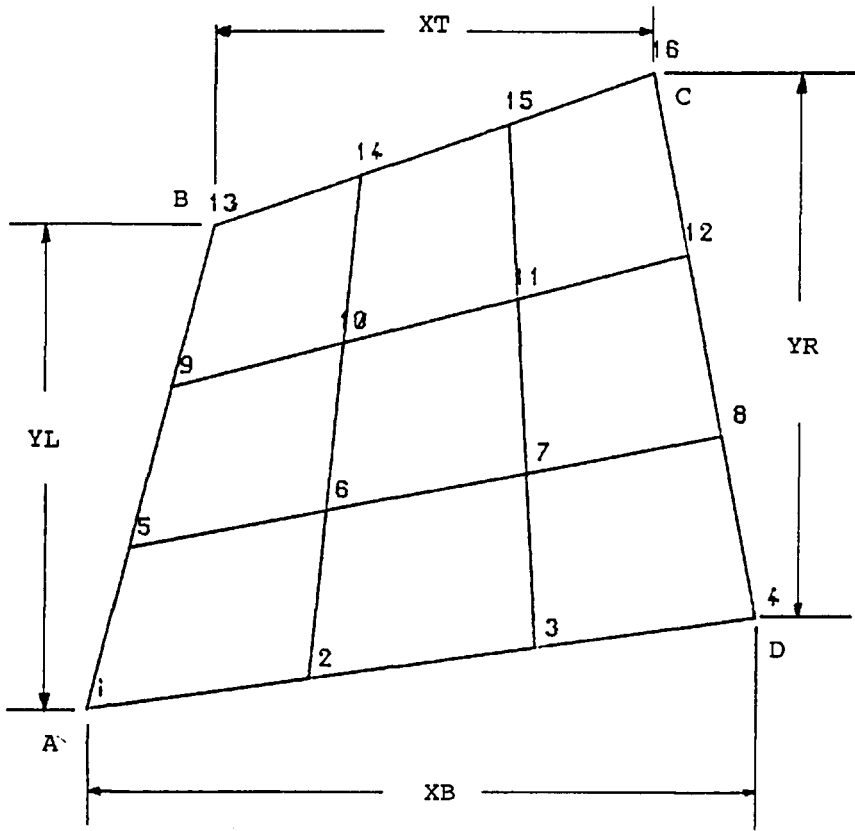


Fig 4.9 A simple quadrilateral mesh

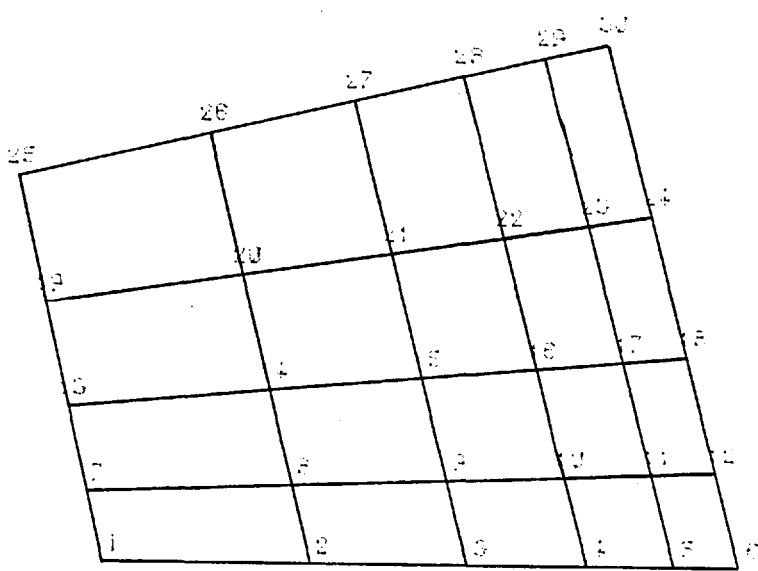


Fig 4.10 Mesh concentration in both X and Y directions

By: rewriting $X(IP)$ of equation 4.14 in terms of $CX(B)$ and $CX(C)$, substituting 4.15 and 4.16 into 4.14 and simplifying we obtain:

$$X(I) = CX(A) + XIR * XB + \{CX(B) + XIR * XT - XIR * XB - CX(A)\} YIR \quad (4.17)$$

Similarly

$$Y(I) = CY(A) + YIR * YL + \{CY(D) + YIR * YR - YIR * YL - CY(A)\} XIR \quad (4.18)$$

Equations 4.17 and 4.18 are general for every nodal point whether it is on the boundary or within the quadrilateral, and for programming purposes only thirteen lines of coding are necessary to generate the coordinates of any node using the definition of the corners and the node number from which the ratios are calculated and hence the coordinates. Listing 4.1 in Appendix C shows the program section that generates the coordinates.

4.1.2 Mesh Concentration

Equations 4.17 and 4.18 used in the automatic generation of the coordinates of any nodal point within the quadrilateral, are based on uniformly distributed nodes on each side of the quadrilateral. There will be cases, where the necessity to concentrate the meshes in certain areas of the model as a result of stress concentration, would be inevitable, thereby creating finer meshes at these regions of stress concentration, and coarse meshes further away from them.

A mesh concentration factor in both X and Y directions may be specified to permit the concentration of meshes along any boundary of the quadrilateral. Fig 4.10 shows a quadrilateral in which the elements of the mesh are concentrated in two directions by utilizing a concentration factor Cx in the X - direction and Cy in the Y - direction.

The values responsible for the even distribution of the nodes in equation 4.17 and 4.18 are the ratios XIR and YIR. These parameters must be redefined to accommodate the change required for concentrating the meshes. They become:

$$XIR_C = \frac{C_x^{(IX-1)} - 1}{C_x^{(NX-1)} - 1} \quad (4.19)$$

$$YIR_C = \frac{C_y^{(IY-1)} - 1}{C_y^{(XY-1)} - 1} \quad (4.20)$$

where Cx is the mesh concentration factor in the X direction

Cy is the concentration factor in the Y direction

and the suffix C in XIR_C and YIR_C is used to differentiate these ratios from those of equations 4.10 and 4.11.

By inspecting the last two equations it can easily be seen that when Cx or Cy become unity the value of XIR_C or YIR_C respectively will remain unchanged whatever the value of IX or IY as the case may be. Thus limits are set for Cx and Cy as follows:

$$0 < C_x < 1 \text{ or } C_x > 1$$

$$0 < C_y < 1 \text{ or } C_y > 1$$

Replacing X_{IR} and Y_{IR} with X_{IR_c} and Y_{IR_c} respectively in equation 4.17 and 4.18 will produce the equations necessary for generating the concentrated meshes. To avoid the meaningless situation when any of the mesh concentration factors is unity, the program uses the definition of X_{IR} and Y_{IR} in equations 4.10 and 4.11 when this condition arises. Some typical meshes with various concentration factors are shown in Fig 4.11. A listing of the program section used for generating the coordinates for both evenly distributed and concentrated meshes is shown in List 4.2.

Using this new algorithm, an overlay called MESH2D has been developed to produce both evenly distributed and concentrated meshes. A flow chart of MESH2D is shown in Fig A.1.1 Appendix A.

4.2 JOINING THE QUADRILATERALS

Using the overlay MESH2D any quadrilateral can be created and subdivided into a mesh by having all its sides or part of its sides joined to existing quadrilaterals to produce a unique mesh. The ability to find points on the existing part of the model and uniquely defining the nodal points that are common to the new quadrilateral being created and the existing drawing, are achieved both by hardware and software. With the aid of the digitising pen and the visual display unit,

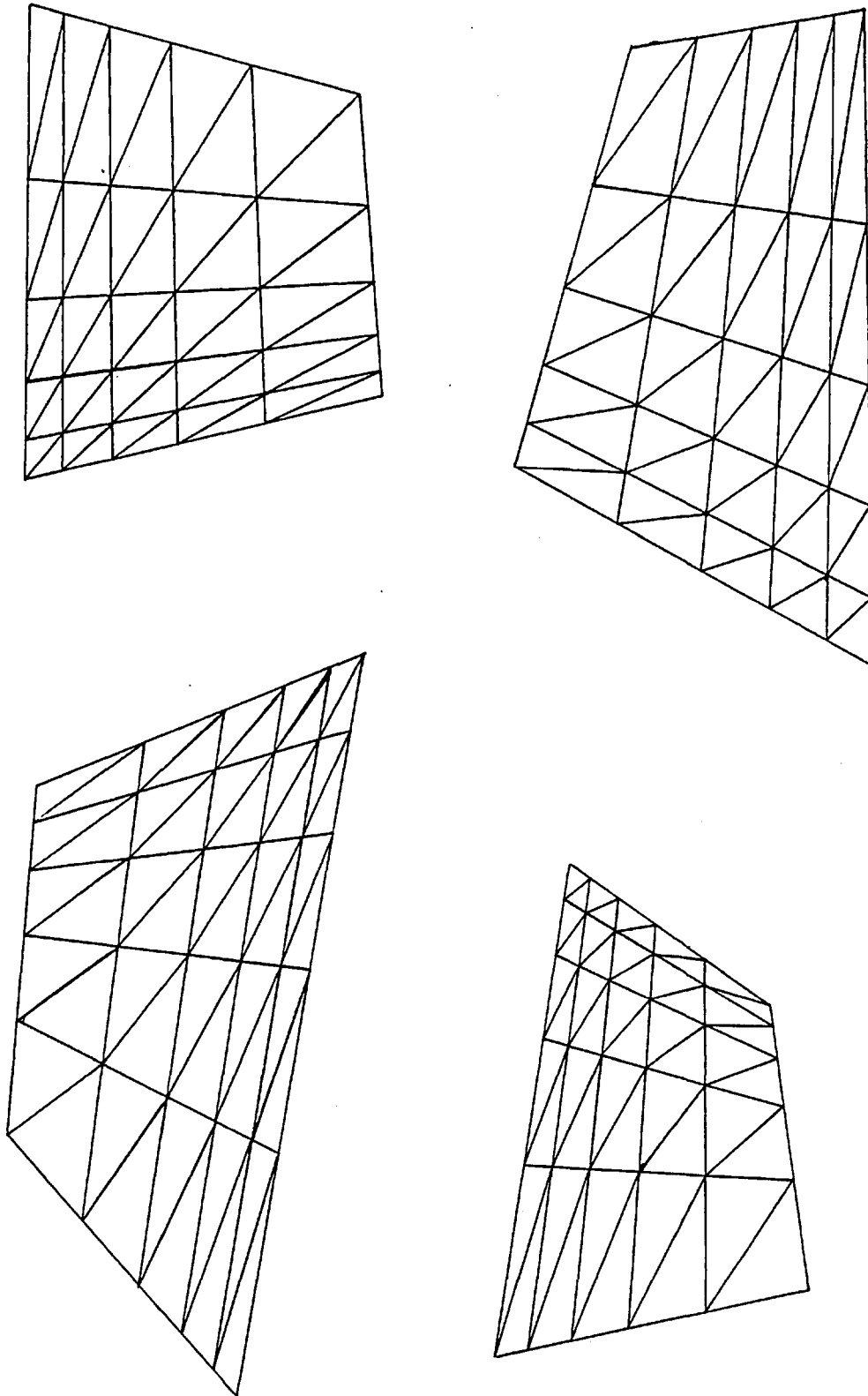


Fig 4.11 Different mesh concentration plots

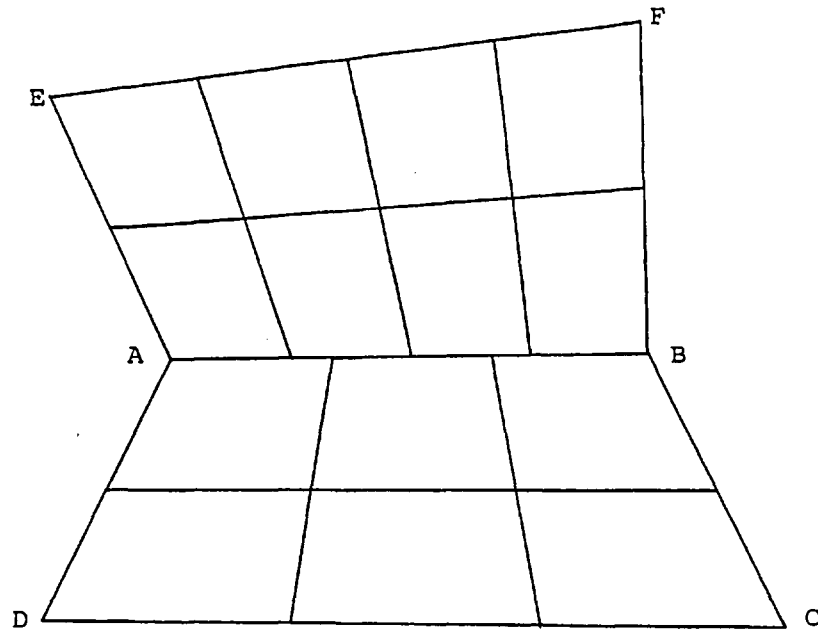


Fig 4.12 Wrong Specification of number of elements in adjacent quadrilaterals

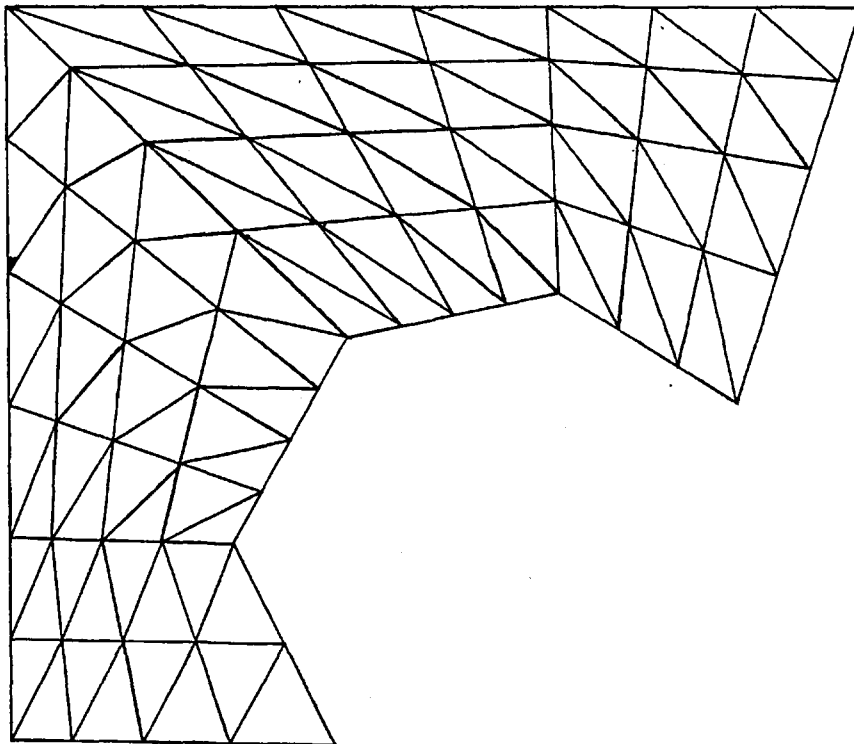


Fig 4.13 Accurately joined quadrilaterals

the user is capable of digitising a point within a maximum distance of five millimetres of an existing point using button 7 of the digitising pen. Then the Find Mode in the basic system software would obtain the exact coordinates of the required point by trapping the nearest point in the workspace to that digitised.

When the full length of any particular side of the quadrilateral is made common to an existing model, the number of nodal points or elements on that side is dictated by that of the existing model. It becomes the responsibility of the user to make sure that he is not creating different number of elements on the side of the new quadrilateral from that of the existing side common to the model and the new quadrilateral, as in Fig 4.12 where the new quadrilateral AEFB has five nodes on line AB which on the previously generated quadrilateral ABCD has only four nodes. Again, when part of a side of the new quadrilateral is being created as common to the existing model, the user should input the number of elements on this side concerned such that the coordinates of the common nodal points will not be doubly defined.

It is also important for the user to know the numbering of the nodal points sequence in any quadrilateral mesh while joining it to an existing figure. The numbering sequence of the nodal points depend on the direction of digitising the four corners of the quadrilateral (clockwise or anticlockwise), and the first node number supplied by the user.

On meshing a new quadrilateral, the nodal points are temporarily numbered from one to $NX * NY$ (where NX and NY are as defined in section 4.1.1); starting from the first corner digitised along the line joining the first and last corners digitised. This line, no matter its direction, is called the X - line and the line between the first and second points digitised is called the Y - line. The first node number on the permanent numbering of the nodes, will be supplied by the user and it is usually taken as the last node number in the previous mesh, incremented by one. Since it is possible to have any side of the new quadrilateral from existing lines in the work space, it is essential to know the temporary numbers of the nodes generated on the new quadrilateral that will lie on the boundary common to the figure already produced in the work space. These node numbers will be specified by the user, so that the subroutine responsible for producing the final node numbering will not renumber them after the first node number is specified. It is very convenient to digitise a new quadrilateral being joined to another in such a way that the X - line coincides with the common boundary, for by so doing, the common nodes will be:

1, 2, 3, ... , $NX - 1$, NX

Specifying these nodes as common is a lot easier for the user since the input is performed manually using the buttons of the digitising pen. Should the user digitise the corners such that the Y - line lies on the common boundary then the common

nodes will be:

$$1, \quad NX + 1, \quad 2NX + 1, \quad , \quad (NY-1) NX + 1$$

For the line opposite the X - line lying on the boundary, the common nodes are

$$(NY - 1) * NX + 1, \quad (NY - 1) * NX + 2, \quad (NY - 1) * NX + 3$$

$$. NX.NY$$

Finally for the line opposite the Y - line lying on the boundary, the common nodes are

$$NX, \quad 2NX, \quad 3NX, \quad (NY - 1)NX, \quad NX.NY$$

To assist the user identify the particular line on the boundary, the line on which the numbering starts (the X - line) will be shown on the VDU.

Since MESH2D is a general program for generating meshes in quadrilaterals, its defined steps of operation are fixed and the user is asked for common nodes even on the first quadrilateral digitised, and inputting zero at these points will advance the sequence of operation, which is another input required by the program, the first node number. The program will then give all the nodes but the ones specified as common, a new node number starting from the first node number, and creates a coordinate data file if required by the user.

The interactive nature of the program requires the user to specify a file number for the coordinate data file, and the user must specify a file number that has not been created already. This is very important since specifying a file number that has been created will cause a crash of the system, besides the complete joining of all the quadrilaterals to form the desired shape takes place in two stages: the graphic stage that has just been described above, and the file manipulation stage in which all the separate data files are joined to form the data file of the complete model.

4.3 MESH DATA STORAGE AND FILE MANIPULATION

Two specific methods are usually available for presenting numerical data to any finite element program. The first involves the use of text cards in the form of punched cards and this is currently the most popular and widely used method. The second method involves the extraction of the numerical data from a magnetic tape in binary format, where the data on magnetic tape is usually generated from a preprocessor provided by the user-such as SAMG. Primarily SAMG will generate numerical data such as nodal points coordinates, element connections, boundary conditions, material properties and applied loads for 2D finite element programs. The numerical data is stored on disk as formatted card images in permanent files, thus facilitating the production of punched cards from these numerical data. The permanent files can be varied in size and number and are only restricted by the size of the disk. These files must all be numbered and named for

identification purposes, and no two files can be given the same name and number. The naming of the permanent files is transparent to the user and it is done automatically by calling either of two subroutines: CONECT or STORE, using the following Fortran statements.

```
CALL CONECT (IFILE)
CALL STORE  (IFILE)
```

where IFILE is the file number to be opened for an input/output operation and resident on disk.

The subroutine CONECT will create the element connection file:

```
C IFILE.DAT
```

and STORE will create the coordinates data file

```
M IFILE.DAT
```

where in both cases C, M and the extension DAT are unchanged arguments of the file names and

```
1 ≤ IFILE ≤ 32767
```

As an example, the statement CALL STORE(50) will create a file named 'M50.DAT' on the system disk. Similarly the statement CALL CONECT (10) will create a file named 'C10.DAT'. These files can be accessed by standard Fortran READ/WRITE statements or by the use of a Fortran subroutine 'SETFIL' which overrides default values for a logical unit at run time.

The coordinate data files are universal and can be used by any finite element program since it contains the nodal points numbers and the corresponding coordinates.

By simply inserting the appropriate control cards this data can be used by every known finite element program. Unlike the coordinate data files, all the element connection files have been written for use with ASKA^[27] and transferability will require a slight change in software. This is caused by the specific orientation of numbering the nodes of each element clockwise or anticlockwise, by different finite element programs. Since there are only two options the problem is not serious for 2D element connection data.

The user is able to edit these permanent files by using the standard DOS/BATCH EDITOR or by using an overlay 'MOVPT'. The latter will move the position of any nodal point and replace the previous coordinates of the point in the permanent file with the new coordinates. The user may also delete these permanent files by using the standard DOS/BATCH peripheral interchange program(PIP) or directly from the digitising table, simply by menu command and specifying the file number. For instance the Fortran statement CALL DELF(500) will delete the permanent file named 'M500.DAT' on the system disk while CALL DELFCN(1000) will delete the file named 'C1000.DAT'. These two Fortran call statements are transparent to the user, but are used by the system software when the user deletes any file by menu command. The user is guided by instruction on the visual display unit to input the file number he wants to delete and to specify whether it is coordinate file, element connection file or both.

Storing the coordinates of the nodal points and the element connections of each quadrilateral in separate permanent files on disk might at first sight seem an inefficient use of secondary storage, and time consuming. But this method of data storage has considerable advantages in that the user may have access to each individual data file for editing. Moreover the data in the permanent files will remain intact when the operation of the system is terminated deliberately or accidentally. When the user is satisfied with all the separate quadrilaterals in a model, the overlay "JOIN" can be used in concatenating these separate data files into one coordinate data file and one element connection file. The overlay "JOIN" will request from the user, by displaying a message on the visual display unit, the file numbers in the order in which they are to be connected to one another, after which the user is again asked for a new number to identify the concatenated file containing the coordinates of all the elements or element connection of the whole model.

At each stage of the data preparation the user is able to transfer some or all of the permanent files from the system disk to any one of the other peripheral devices such as the Tektronics 611 (DS), the line printer (LP), the teletype keyboard (KB), the paper tape (PP), the magnetic tapes (MT0 and MT1), and the data disk (DK1). This can be done mainly for the purpose of saving the permanent files or checking their contents. While the first three devices mentioned above are mainly for checking the contents of the files, the last three are used for saving purposes, thereby allowing the user to

delete some or all of the data files on the system disk to create space for the creation of more files, or for general usage on the system disk.

4.4 2D-MESH EDITING AND AUTOMATIC RENUMBERING OF NODES

In general, there are two methods that can be used in the description of a mesh, and they are Top down and Bottom up. The method described in the automatic mesh generation equations is a Top down method with respect to the individual building blocks - the quadrangles. There will be instances when the graphical shape of the model does not facilitate the use of the automatic mesh generation system. Clearly this will happen when the structure cannot be effectively subdivided into series of quadrilaterals, and in such cases elements can be added, deleted and nodal position adjusted manually - the Bottom up approach. Manual input facilities complementary to the automatic mesh generation technique, are provided to permit the rapid generation of meshes for difficult shapes.

4.4.1 Add Element

The manual input of additional elements is performed by displaying the model on the visual display unit and continuously adding elements to the model until the desired mesh is complete. For example, a triangular mesh would be added by digitising two fixed points on the screen using the FIND mode, and the third point of the triangle would be moved relative to the movement of the digitising pen until the pen is pressed at the required point for the vertex.

The overlay 'ADDELM' responsible for this operation will then find the connection of this new element, and add its connection pattern to the existing data file, also adding the coordinates and number of the node to the coordinate file. It is possible to create new elements using one, two, three or four existing point or points of the model, a maximum of three for triangular elements and four for quadrilateral elements. The user is not permitted to create an additional element that is not connected to at least one existing node in the workspace. The code for the overlay ADDELM is written in such a way as to be in a continuous loop for the addition of as many elements as the user may require and can only be terminated at the request of the user. A flow chart of ADDELM can be seen in Fig A 2.1 of Appendix A.

4.4.2 Delete Element

Sometimes it might be easier to enclose an area with automatic mesh generation and delete some of the elements not required in the shape of the model. For example if a hole is inserted in a plate, it will be easier to generate the mesh for the plate automatically, delete the elements in the region of the hole and rearrange or move the nodes around the hole to lie on the circumference. To delete an element the user is directed by the program to use the FIND mode to input all the nodes of the element to be deleted, which will subsequently be displayed on the VDU for acceptance or rejection by the user. Should the user accept to delete the displayed element, the program deletes the element and instructs the user to input the

corners of the next element he wants to delete or exit. When the user requests for exit, the program does some housekeeping and tidying up of the data files in a mode transparent to the user before relinquishing control to the User Operating Monitor 'DIGOV'.

4.4.3 Adjust Position of Node

The movement of the position of a node or nodes and hence the position of an element without changing the element connection nor the total number of elements and nodes is essential as can be seen from the example of a plate with a hole. This editing facility is also available in the system for the user to digitise the node to be moved using the FIND mode, whence the movement of the node will be controlled by that of the digitising pen and its new position entered by pressing button One on the digitising pen. The old coordinates of the node are replaced by the new coordinates in the co-ordinates' data file.

4.4.4 Automatic Renumbering

For the solution of sets of linear simultaneous equation, the most commonly used techniques are the direct Gaussian elimination and the iterative Gauss-Seidel methods which have the advantage of speed and minimum computer core requirement. For Finite Element Methods, the matrix equations are symmetrical and all the non-zero coefficients are confined within a band whose width is calculated as the maximum difference between any two related nodes, incremented by one

for the diagonal terms. To take the fullest advantage of the benefits of the above solution methods, the programmer must make sure that the matrix bandwidth is as narrow as possible. Automatic node renumbering to reduce the bandwidth can reduce both the storage requirements and solution time for a set of linear simultaneous equations whose form is a function of the interconnection of the nodes in a structural frame work. Although there are many different programs used in the reduction of bandwidth, the overlay 'OPBAND' uses an algorithm based on Collins [22] algorithm, and modified by Ghassemi [10]. A detailed description of the program OPBAND and the algorithm can be obtained from Ghassemi [10].

4.5 DOUBLE AND ROTATE

The production of meshes for axisymmetric shapes is achieved by the use of two main techniques: Double and Rotate. In either case a full 2D mesh is first created and stored as described earlier in the chapter. For Double, once a planar mesh is generated, it could be projected as many times as required into other planes parallel to the first, thus producing a three dimensional kind of mesh by joining the corresponding points on any two adjacent planes of projection. The size of the new plane projected can differ from that of the original plane. For Rotate, the original mesh is rotated about an axis parallel to the Y axis, and the number of planes can be specified by the user to control the number of 3-D elements produced. Both methods produce 8-Node brick elements or six node wedge elements.

4.5.1 Theory

a) Double

The first plane containing the 2-D mesh is usually the X - Y plane, and any subsequent projection will be parallel to the first controlled by three variables: Δx , Δy and Δz such that if X_i, Y_i and Z_i are the coordinates of a point on the old plane being projected and x_i, y_i and z_i are the coordinates of the corresponding point on the projection plane, then

$$\begin{aligned} x_i &= X_i + \Delta x \\ y_i &= Y_i + \Delta y \\ z_i &= Z_i + \Delta z \end{aligned} \quad (4.21)$$

Equation 4.21 can only give projection plane equal in all respects to the projected plane, only shifting the coordinates as determined by the shift variables Δx , Δy and Δz . To change the size of the projection plane a fourth control variable (the projection factor F) is required. For repeating the projections a fifth variable (the number of projections N) is needed.

Define the following:

X_c = The mean X coordinate or the X - coordinate of the centre of figure

Y_c = The mean Y coordinate

To account for the projection factor F ,

Define:

$$X = |X_i - X_c| * F \quad (4.22)$$

$$Y = |Y_i - Y_c| * F \quad (4.23)$$

For X_i greater than X_c

$$x_i = X_c + X + \Delta x \cdot n \quad (4.24)$$

For X_i less than X_c

$$x_i = X_c - X + \Delta x \cdot n \quad (4.25)$$

Rewriting equation 4.24 and 4.25 as a single equation we have

$$x_i = X_c \pm |X_i - X_c| * F + \Delta x * n \quad (4.26)$$

$$\text{similarly } y_i = Y_c \pm |Y_i - Y_c| * F + \Delta y * n \quad (4.27)$$

$$\text{and } z_i = (Z_i + \Delta z) * n$$

where $1 \leq n \leq N$, n being the projection plane number and in equations 4.26 and 4.27 the positive sign prevails when

$$X_i > X_c \quad \text{and} \quad Y_i > Y_c$$

and the negative sign for $X_i < X_c$ and $Y_i < Y_c$

b) Rotate

Again the first plane that contains the 2-D mesh is the $X - Y$ plane, and all subsequent rotation planes are produced such that they are at defined angles from the first plane about the axis of rotation. The definition of the angles depend on the number of rotation planes such that the angle between the two adjacent planes will be the same. Let the number of planes of rotation be N . Then the angle between the two successive planes ΔA is given by the following equation

$$\Delta A = \frac{2\pi}{N}$$

Again let the coordinates of a nodal point in the first plane

be X_i , Y_i and Z_i and those of the corresponding point on any plane of rotation be x_i , y_i , z_i . Since the mesh to be rotated is in the $X - Y$ plane and subsequent plane of rotation are about the Y axis, it is easily seen that the Y - coordinates of all the rotated points will not change, and only the X and Z coordinate are affected by such a rotation. Furthermore since the axis of rotation is on the XY plane some of the planes will have negative Z coordinates, and to avoid that we define a point to the right of the figure to give the Z coordinate of the centreline. In the ZX plane the centreline can be represented by a point. Let the coordinates of this point be X_c, Z_c .

Define the following constants

$$X = X_c - X_i$$

$$Z = Z_c - Z_i$$

From Fig 4.14 it can be seen that by simple rotational matrix equation

$$x_i = X \cos \theta - Z \sin \theta + X_c$$

$$z_i = X \sin \theta + Z \cos \theta + Z_c$$

and since the Y coordinate doesn't change

$$Y_i = Y_i$$

$$\text{where } \theta = \Delta A * \eta$$

and η is the number of the plane of rotation

$$1 \leq \eta \leq N$$

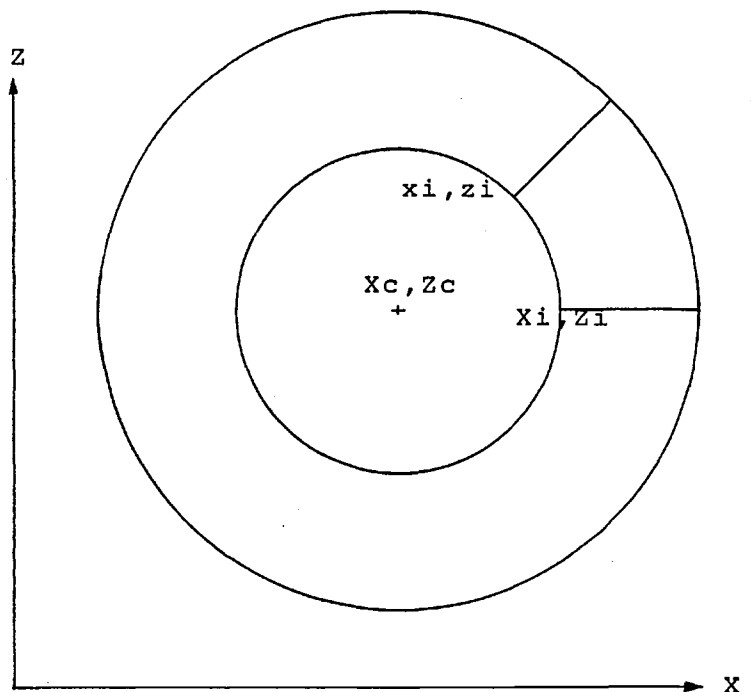


Fig 4.14 2-D Rotation

4.5.2 Nodal Point Coordinates and Element Connection for 2D Files

As stated earlier in the chapter the 2D files from which the axisymmetric shapes are constructed are stored in permanent files on the system disc. These files are modified for three dimensional mesh data by simply adding the co-ordinates and node numbers of the projection or rotation nodes, to the permanent files to represent a three dimensional structure using Fortran READ/WRITE statements. The numbering sequence of the projection planes will be similar to that of the first plane. Each nodal point number on the projection plane will be equal to its corresponding number in the first plane, plus the total number of nodes generated in all the previous planes. Mathematically, the last statement can be thus presented:

$$I = I_{xy} + \eta * NX * NY$$

where I is the new node number in the η^{th} plane

I_{xy} is the number of the node in the corresponding position in the XY plane

η is the projection plane number

NX is the number of nodes on the X - line

NY the total number of nodes on the Y-line

Therefore if the numbering sequence of the nodes in the X - Y plane is as follows

$$1, 2, 3, \dots, NX.NY$$

the numbering in the first projection or rotation plane will be

1 + 1.NX.NY, 2 + 1.NX.NY, 3 + 1.NX.NY, 2NX.NT
on the second plane

1 + 2NX.NY, 2 + 2NX.NY, 3 + 2NX.NY, 3NX.NY
and so on.

A separate overlay is created to generate the element connection file of this three dimensional structure created by using 2½D technique. The overlay ST3D was originally written by Ghassemi and has been considerably modified by the author.

C H A P T E R 5

3-D SHAPE DESCRIPTION AND MESH GENERATION

In 2-D mesh generation described in the previous chapter, little or no problem is encountered in the definition of the 2D shapes, since the basic input device, the digitising table can convert any point digitised into X and Y coordinates. Three different methods of using the conventional digitiser to input three dimensional data to the computer, are fully described by Yi et al [25]. The first method is based on digitising two or more orthogonal projections which are subsequently assembled within the computer to give the required three dimensional shape. The second method consists of digitising contours in two dimensions with the third dimension normal to the surface, being added numerically, after which a surface patch technique is utilized to interpolate the contour data, in order to produce data of sufficient quality for the graphical representation. Finally, the third method is based on photogrammetric technique where two perspective drawings or photographs are digitised and the data converted into a three dimensional format.

These three methods briefly outlined above, are all very useful if the model under consideration is relatively simple or the storage capacity of the core of the computer is very large. For more complicated shapes and models to be fully represented using a mini computer, a slight modification of the first method is considered in this work, where the model

is divided into basic building blocks and assembled later. The use of building blocks which are later assembled to produce the required shape might seem tedious and time consuming from the shape description point of view, but when considered from a mesh generation point of view, it is the most efficient mode when using a computer of small core store. Each building block can be subdivided into a mesh, and the data thus generated stored in separate files that can be edited. Unlike 2-D mesh generation, different types of building blocks are designed to produce different types of elements, and the method of input for each building block is specifically defined and must be strictly adhered to.

5.1 Input Methods and the MACROBLOCKS

There are six different types of input shapes designed to generate four different types of building blocks called MACROBLOCKS. All inputs are handled and serviced by an interactive overlay called 3D MESH. When this overlay is loaded into core by digitising above the appropriate square on the menu table, the system requests for the number of input points, for the different types of input shapes are recognised by the difference in the number of input points. Therefore it is important for the user to know the type of input required and the number of input points for the desired input shape. The order in which the points of the input shape are digitised is also rigidly defined and on providing the number of input points, '3D MESH' will proceed to guide user to digitise one numbered input point after the other in either the X - Y

plane or the ZX plane. Due to the difficulty in representing a three-dimensional object on a plane surface, each 3-D point is defined by two points on the digitising surface: the first point in the X - Y plane and the second in either the Z - X plane or the Y - Z plane. From these two points the cartesian X, Y and Z coordinates of the three dimensional point are derived by a subroutine called 'CONVRT'.

Since the points on any input shape are specifically numbered, the program using overlay 3D MESH is capable of directing the user to input point number 'N' in the X - Y plane and Z - X plane, 'N' starting from unity to the specified number of input points, in steps of one. After accepting the last input point, the program enters an editor mode when the user is permitted to change the coordinates of any of the input points. The user is requested to specify the number of the input point he wants to change, and is directed to digitise the points in the planes as the program will accept. The rigid definition of the order of entry of the points in any of the six input shapes, compels us to consider the six input shaped individually.

a. Straight edged block with uniform thickness.

In this case only five input points are required to fully define the shape although there are eight corners. The problem of digitising two planar points to represent one 3-D point encourages

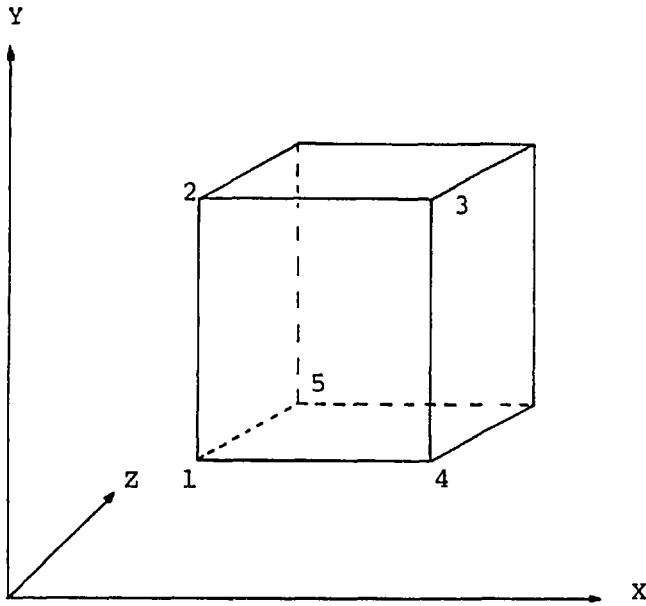


Fig 5.1

cuts in the number of input points where the cut will not affect the accuracy in the overall definition of the shape. This input shape is used by the first two types of building blocks: MACROBLOCK 1 and MACROBLOCK 2. These building blocks are used in generating 8 -

NODE brick elements and 20-NODE isoparametric elements respectively. The first parameter supplied to the overlay 3D MESH is 5 for this input shape, and the second parameter, the macroblock number must be 1 or 2, for the program to load the appropriate meshing overlay into core.

b. The second type of input shape is one in which a planar view has curved edges, but with uniform thickness, Fig 5.2.

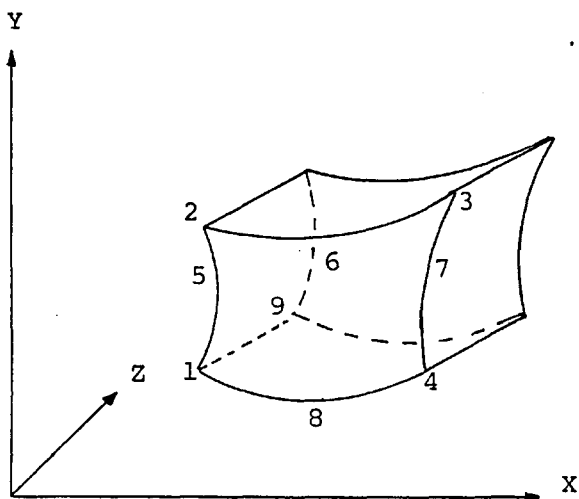


Fig 5.2

In this case the orthographic views of the structure will be taken such that the surface with the curved edge must be digitised in the X - Y plane. Only nine input points are necessary for full definition of the

shape, and must be digitised in the order shown in Fig 5.2. Again this type of input shape is only useful for the first two types of building block, but in practice it is better utilised by the second type of building block which maintains a better approximation to the curve along the edges.

c. The third type of input is in all respect but one, like the first. The difference is only in the number of input points. Instead of five, eight input points are required for this irregular block with straight edges. Theoretically, input shape (a) is a special case of this type of input where only one depth

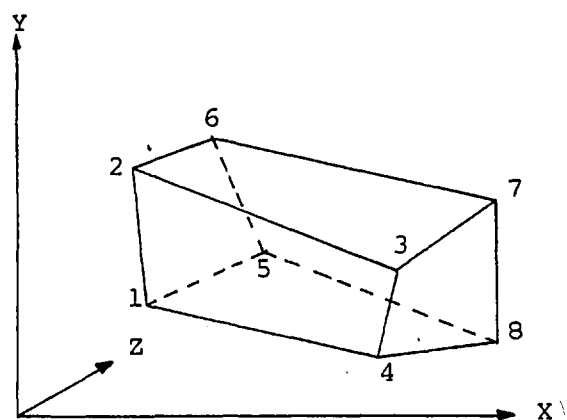


Fig 5.3

or thickness is required by the program to compute the coordinates of ^{the}sixth, seventh and eighth corners. The node numbering routines are responsible for obtaining and using these coordinates of the box. They will be seen later.

d. The fourth type of input shape with twenty points is used as the second type of input shape (b), the latter being a

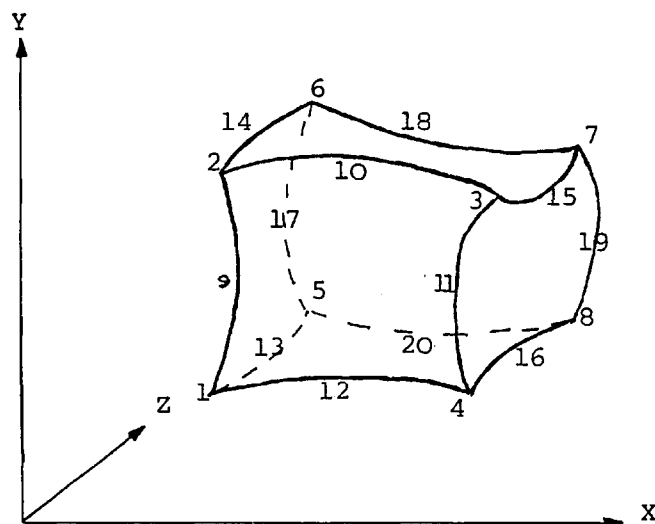


Fig 5.4

special case of this type of input. '3DMESH' will guide the user by requesting for specific input number first in the X - Y plane and then the Z-X plane. It must be emphasized again, that due to the complexity of the nature of the routines involved in the generation of the node numbering of the elements nodes, preparing the shape function parameters of the node, and finally generating the nodal point coordinates, the order of input should strictly be followed to avoid spurious results. This input shape can be used by the first two building blocks but is more useful for MACROBLOCK 2.

e. For solids of revolution whose cross sections have straight edges, the fifth input shape is designed. Six input points are necessary to give a complete definition of such figures. The

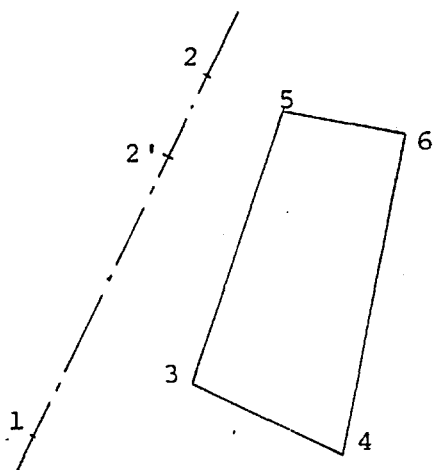


Fig 5.5

first two points define the centre line and the last four points define the section to be revolved about the centre line. The relative positions of the first two points are not critical as long as they lie on the path of the centre line as in Fig 5.5, the second point can be at

2 or 2'. The centre line can have a three-dimensional skew, and the three direction cosines are computed internally by the mesh generating routine. This type of input shape is used by the third^{and} fourth building blocks: MACROBLOCK 3 and MACROBLOCK 4.

These building blocks are used in generating 8-Node and 20 Node cylindrical elements respectively.

f. The last input shape so far is also designed for solids of revolution, and only differs from the last one by having ten 3-D input points. The first six points are like those defined in the last input shape, which take care of the definition of the centre line and the four corners of the shape to be re-

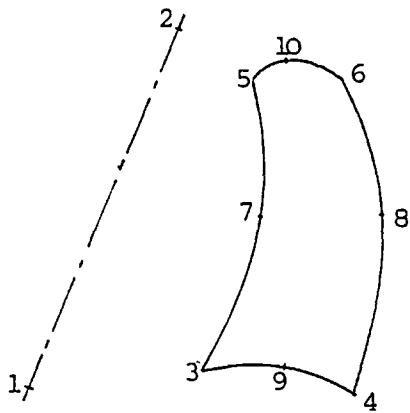


Fig 5.6

volved. The last four points define the mid points on the boundary lines of the shape of revolution. Thus introducing a parametric curve on the lines of the shape to be revolved. Like that of the last input shape, the revolution is done in accordance with the right

hand cork screw rule with the direction of the centre line from point 1 to point 2. This input block is most suitable for MACROBLOCK 4 although it can be used by MACROBLOCK 3.

So far only two groups of input have been discussed first the numerical input of the number of the input points to be digitised, and secondly the actual points digitised on the table. Once the number of 3-D points digitised on the table, is equal to that specified by the user the program jumps out of the digitiser input loop and request the user to certify the acceptability of these stored points. If any of the points

are not acceptable to the user, for instance when the program requested for point number three in the XY plane and the user digitised what should be point number five in the particular input shape, the program jumps into an editor phase, guiding the user to correct the point which has been digitised in error. At this stage, all the input points can be changed and if the initial number of input points specified is less than that required, more points could be added. The program can only get out of the editor mode when the user requests for a null point to be edited. Once the user is satisfied with the input points, the program requests for the macroblock number, to call the right overlay into core for the building block required by the user.

Before loading the appropriate service overlay for the required building block into core 3DMESH stacks its third entry point on the overlay stack, so that control of the system is returned to it after the mesh is developed by the service routine. All the routines used for the generation of meshes within the separate building blocks are written in such a way that if the number of input points specified at first entry to 3DMESH is not compatible with the routine, control is automatically handed over to 3DMESH at the second entry point where the user can correctly input the required macroblock. The user can exit from 3DMESH by supplying a macroblock number of six - a dummy macroblock number used for exit purposes. When a mesh generating routine is successfully executed and control handed over to 3DMESH, the element connection generating routine is loaded only if a coordinate file is stored on the system disk. After

the element connections are generated and a file created, or if there is no coordinate file 3DMESH will ask the user for more macroblocks or an exit from the program. If the addition of more building blocks to the workspace is requested, 3DMESH places its first entry point on the overlay stack and calls the overlay responsible for displaying the three orthogonal views 'DVIEW'. After displaying the three orthogonal views, '3DMESH' is loaded into core and the first request from the program 'NUMBER OF INPUT POINTS' appears on the VDU. Fig A3.1 in Appendix A shows a flow chart of 3DMESH.

The use of the overlay 3DMESH as a coordinating program for the whole 3-D mesh generation cannot be overemphasised. It is responsible for loading and coordinating four separate mesh generation overlays, an element connection data generating overlay, and last but most important itself.

5.2 MESHING THEORIES

There are three separate theories involved in the meshing of the four kinds of building blocks:

- a. Coordinate geometric scheme
- b. Isoparametric mapping
- c. Matrix transformation

The three schemes are either used separately or combined to produce the required mesh in the different macroblocks.

5.2.1 Coordinate Geometry Algorithm 3-D

In the previous chapter, section 4.1, it was discussed that the desire to change the commonly used technique of mesh modification from that of a unit square, was to make it easier to develop a similar three dimensional theory. Thus by knowing the eight sets of coordinates for the corners of a box we should be able to extrapolate some of the equations in Chapter 4, and arrive at a suitable set of equations to do the same thing in three dimensions.

Consider the box 1-2-3-4-5-6-7-8 in Fig 5.7, which is divided into four smaller brick elements. Line 1-4, no matter its direction is termed the X line, line 1-2 is the Y line and line 1-5 is the Z line. The numbering of the nodal points is such that it starts from corner 1 to corner 4 before the next row on the surface 1-2-3-4, where all the nodes are numbered before the numbering starts on the next plane of nodes in the increasing direction of the Z line. Each plane (actually surface) of nodes is numbered in exactly the same order as the two dimensional numbering sequence in Chapter 4. Thus if we define IX, IY and IZ as counters along the primary lines, the typical node number can be derived from the following equation:

$$I = (IZ - 1) NY * NX + (IY - 1) NX + IX \quad (5.1)$$

where I is the number of the node

$$1 \leq IZ \leq NZ$$

$$1 \leq IY \leq NY$$

$$1 \leq IX \leq NX$$

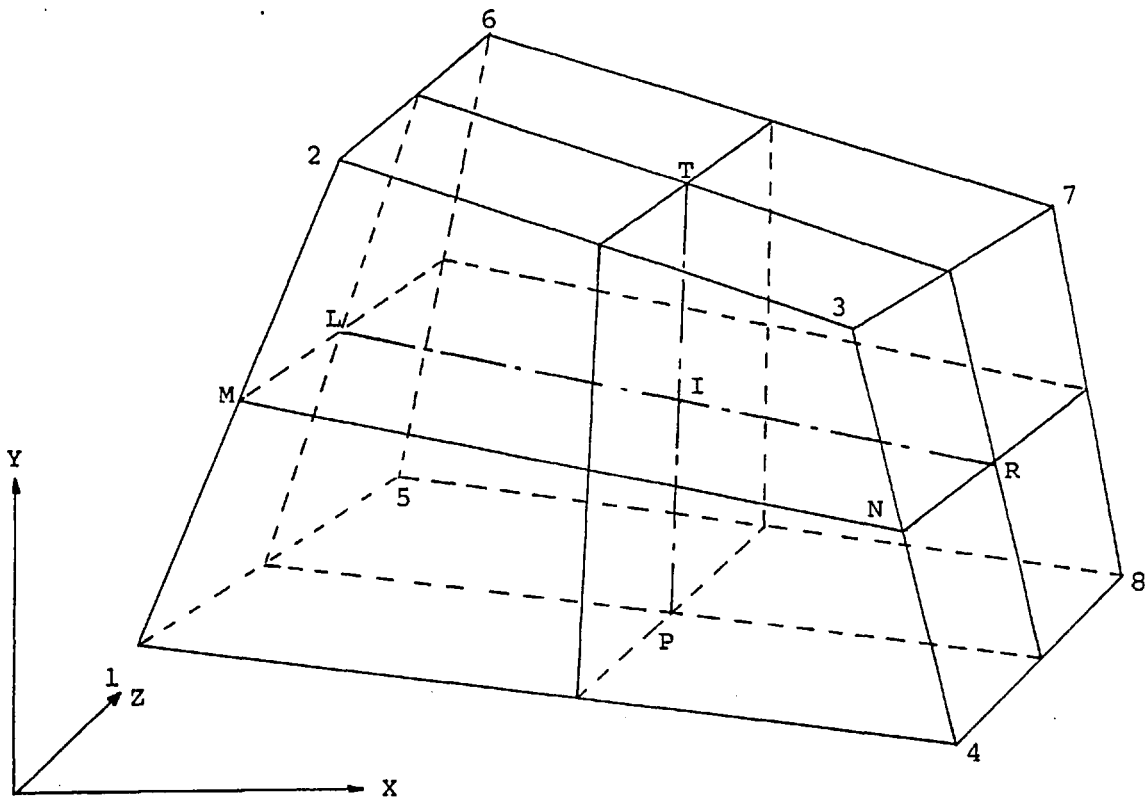


Fig 5.7 Actual Block

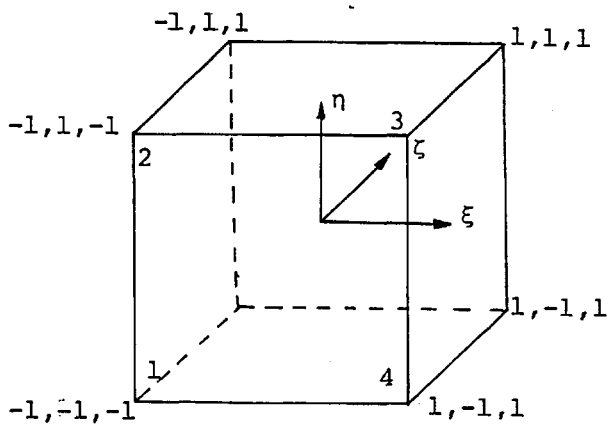


Fig 5.8 Scaled block

NZ is the total number of nodes on the Z line
 NY is the total number of nodes on the Y line
 and NX is the total number of nodes on the X line

Similar to the two dimensional case, let XIR, YIR and ZIR be the ratio of the nodal point position in the three principal directions. Then these ratios for each nodal point are obtained from the following equations:

$$\begin{aligned} XIR &= (IX - 1) / NXL \\ YIR &= (IY - 1) / NYL \\ ZIR &= (IZ - 1) / NZL \end{aligned} \quad (5.2)$$

where $NXL = NX - 1$ (The number of elements along the X line)
 $NYL = NY - 1$ (The number of elements along the Y line)
 and $NZL = NZ - 1$ (The number of elements along the Z line)

Consider the nodal point I in Fig 5.7

$$\begin{aligned} IX &= IY = IZ = 2 \\ X(I) &= X(P) + \{X(T) - X(P)\} * YIR \end{aligned} \quad (5.3)$$

where $X(N)$ is the X coordinate of point N

From equation 4.16 $X(P)$ can be expressed in terms of XIR, ZIR and the X coordinates of the points 1,4,8 and 5. Similarly $X(T)$ can be expressed in terms of the two ratios above and the X coordinates of points 2,3,7 and 6.

$$X(P) = X(1) + XIR\{X(4) - X(1)\} + \{X(5) + XIR [X(8) - X(5)] - [X(1) + XIR(X(4) - X(1))]\} * ZIR \quad (5.4)$$

$$X(T) = X(2) + XIR\{X(3) - X(2)\} + \{X(6) + XIR [X(7) - X(6)] - [X(2) + XIR(X(3) - X(2))]\} * ZIR \quad (5.5)$$

$$\begin{aligned} \text{Let } XB &= X(4) - X(1) \\ XC &= X(3) - X(2) \\ XD &= X(2) - X(1) \\ XE &= X(5) - X(1) \\ XF &= X(8) - X(5) \\ XG &= X(7) - X(6) \\ \text{and } XH &= X(6) - X(2) \end{aligned}$$

Then

$$X(P) = X(1) + XIR*XB + ZIR*XE + XIR*ZIR(XF - XB) \quad (5.6)$$

$$X(T) = X(2) + XIR*XC + ZIR * XH + XIR*ZIR(XG - XC) \quad (5.7)$$

Similarly by defining

$$\begin{aligned} YB &= Y(4) - Y(1) \\ YC &= Y(3) - Y(2) \\ YD &= Y(2) - Y(1) \\ YE &= Y(5) - Y(1) \\ YF &= Y(8) - Y(5) \\ YG &= Y(7) - Y(6) \\ \text{and } YH &= Y(6) - Y(2) \end{aligned}$$

$$Y(L) = Y(1) + YIR*YD + ZIR*YE + YIR*ZIR(YH - YD) \quad (5.8)$$

$$Y(R) = Y(4) + YIR*YC + ZIR*YF + YIR*ZIR(YG - YC) \quad (5.9)$$

$$Y(I) = Y(L) + \{Y(R) - Y(L)\} * XIR \quad (5.10)$$

The Z coordinate of the point I can be obtained from either of two equations:

$$Z(I) = Z(L) + \{Z(R) - Z(L)\} * X_{LR} \quad (5.11)$$

$$\text{or } Z(I) = Z(P) + \{Z(T) - Z(P)\} * Y_{LR} \quad (5.12)$$

By expressing $Z(L)$, $Z(R)$, $Z(P)$ and $Z(T)$ in terms of the Z coordinates of the eight corner, and fully expanding the last two equations, the same result is obtained for $Z(I)$.

For programming purposes the constant values X_B , X_C , etc, Y_B , Y_C etc and Z_B , Z_C etc are defined outside the coordinates generating loop within which we calculate: the node number I, the ratios X_{IR} , Y_{IR} and Z_{IR} , and the six surface coordinates $X(P)$, $X(T)$, $Y(L)$, $Y(R)$, $Z(L)$, and $Z(R)$. By substituting the last six values into equations 5.3, 5.10 and 5.11 respectively, we generate the X,Y and Z coordinates of any node.

This algorithm is simple and straight forward for generating 8-node brick elements and it is used in the overlay "BLK8ND", which is loaded into core for production of the mesh when the user requests for MACROBLOCK 1. This was the first meshing algorithm developed by the author, but the inadequacy of this method in dealing with building blocks having curved surfaces gave rise to the use of isoparametric mapping techniques.

5.2.2 Isoparametric Mapping

The basic scheme is the use of parametric curvilinear mapping of quadrilateral^s_k in two dimensions and hexahedrals in three dimensions. The scheme, initially developed by Ergatoudis et al 1968^[9] for derivation of special element forms, allows the unique coordinate mapping of curvilinear and cartesian

coordinates. Isoparametric transformations are assumed, relating the actual geometry of a building block to a scaled block. From the unique mapping of the actual cartesian coordinates to the scaled coordinates obtained from these transformations, and the number of elements required in the building block, an interpolation is carried out on the scaled block and related to the actual building block to determine the nodal geometry. To illustrate the method, consider the three dimensional mesh of Fig 5.7 where all the coordinates of the eight corners are given. The transformation of the (X,Y,Z) cartesian coordinates to the scaled coordinates of (ξ,η,ζ) of Fig 5.8 can be written in the general form (as shown by Ergatoudis et al).

$$U = \alpha_1 + \alpha_2 \xi + \alpha_3 \eta + \alpha_4 \zeta + \alpha_5 \xi\eta + \alpha_6 \eta\zeta + \alpha_7 \zeta\xi + \alpha_8 \xi\eta\zeta$$

where U is the displacement component (X , Y or Z) (5.13)

By substituting in the equation 5.13 the actual values of U and the corresponding scaled coordinates at the eight corners, the coefficients α can be found and the equation written in another general form:

$$U = [N_1, N_2, \dots, N_8] \begin{bmatrix} U_1 \\ U_2 \\ \vdots \\ U_8 \end{bmatrix} \quad (5.14)$$

or $U = \sum_{i=1}^8 N_i U_i$

where N_i is called shape function and is associated with each node and defined in terms of the curvi-linear coordinate system ξ, η and ζ each of which has a value ranging from -1 to 1 on the opposite sides. Typical shape functions are given in general form for a variety of elements by Zienkiewicz et al [26]. In our example there are eight shape functions.

$$\begin{aligned} N_1 &= \frac{1}{8} (1 - \xi)(1 - \eta)(1 - \zeta) \\ N_2 &= \frac{1}{8} (1 - \xi)(1 + \eta)(1 - \zeta) \text{ etc.} \end{aligned}$$

Rewriting equation 5.14 only for the X coordinates, the general form will be

$$x = [N_1, N_2 \dots] \begin{bmatrix} X_1 \\ X_2 \\ \vdots \end{bmatrix} \quad (5.15)$$

where X_n is the X coordinate at corner n
and $N_i \equiv N_i(\xi, \eta, \zeta)$

Equation 5.15 gives a unique mapping of the actual building block to the scaled block, and the coordinates of Y and Z can be similarly described. Quite obviously, we need to know the ξ, η and ζ coordinates of every nodal point on the scaled block to obtain the actual coordinates by substituting the values of (ξ, η, ζ) into equation 5.15. These scaled coordinates values are computed by the node numbering routine which will be seen later. A separate subroutine is then used to compute the values of the shape functions using the scaled coordinate values and finally the actual coordinates are

calculated using the shape functions. This method so far described has been used in generating the mesh in, MACROBLOCK 1 and MACROBLOCK 2, both building blocks using the same shape function generating routine. This explains why the input points for the twenty-point input shape should be such that the first eight points are the corners of the isoparametric box in the same order as the irregular block of input shape (C).

5.2.2.1 Node numbering and Computing Scaled coordinates

To produce 8-node brick elements the node numbering is the same as discussed in section 5.2.1 symbolized in equation 5.1.

$$I = (IZ - 1)(NY)(NX) + (IY - 1)NX + IX \quad (5.15a)$$

In general the scale coordinates can be obtained from the following equation

$$U_i = 2R_i - 1 \quad (5.16)$$

where R_i is the nodal position ratio in the particular direction hence

$$\begin{aligned} \xi_i &= 2.X_{IR}-1 \\ \eta_i &= 2.Y_{IR}-1 \\ \zeta_i &= 2.Z_{IR}-1 \end{aligned} \quad (5.16a)$$

where X_{IR} , Y_{IR} and Z_{IR} are as defined in Equation 5.2

By examining equation 5.16a for the first nodal point at corner 1 of the hexahedron

$$\begin{aligned}\xi_1 &= -1 \\ \eta_1 &= -1 \\ \text{and } \zeta_1 &= -1\end{aligned}$$

These values of the scaled coordinates correspond to those in Fig. 5.8.

For numbering the nodal point on MACROBLOCK 2, a different strategy has to be employed since the number of nodes along any mid point row or column is different from that of the next which forms an edge of any element, along the row or column. To illustrate the above statement, consider Fig 5.9 where there are five nodal points on the line AB but the next row containing the mid points of the first two elements in the plane ABCD, has only three nodes. In fact, a single equation cannot be used in obtaining the scaled coordinates. Three sets of equations were derived to produce a unique numbering system to obtain the scaled coordinates.

Considering Fig 5.9 the line AB, no matter its direction, is called the X line, AC the Y line and AE the Z line. Using the defined values of IX, IY, IZ, NXL, NYL, NZL the numbering of the nodal points was divided into three groups. The first group consists of all the nodes that form corners of the elements, all mid point nodes that lie on lines parallel to the X line in

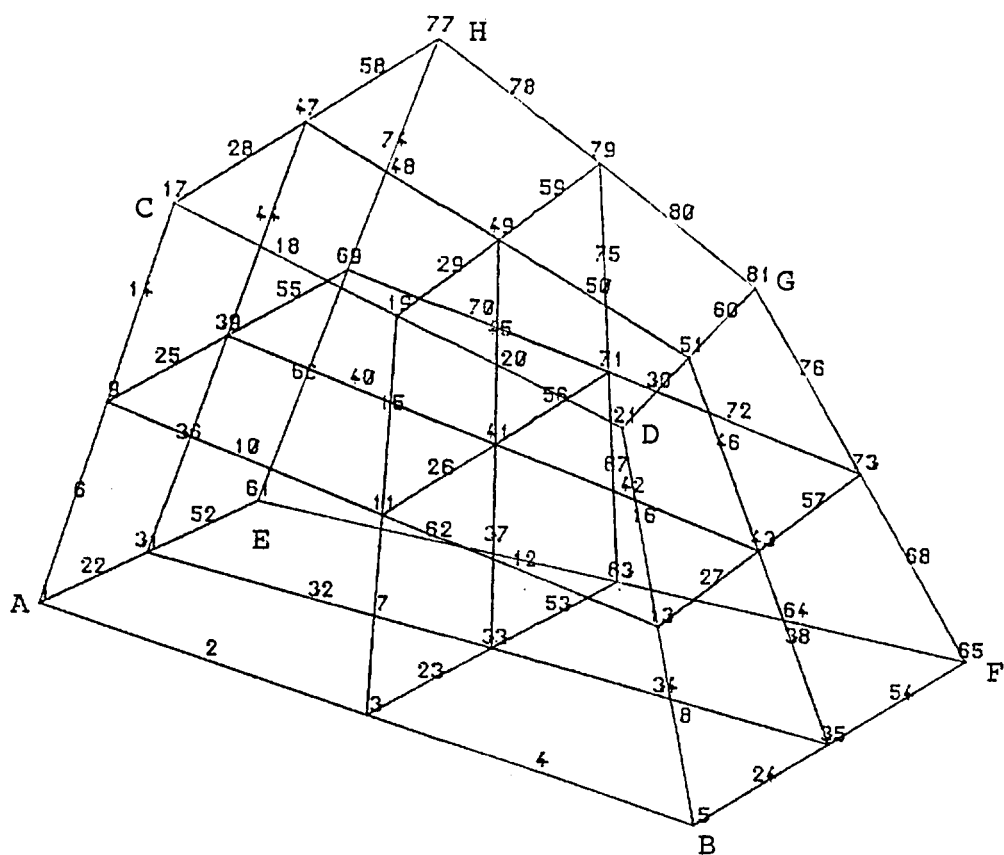
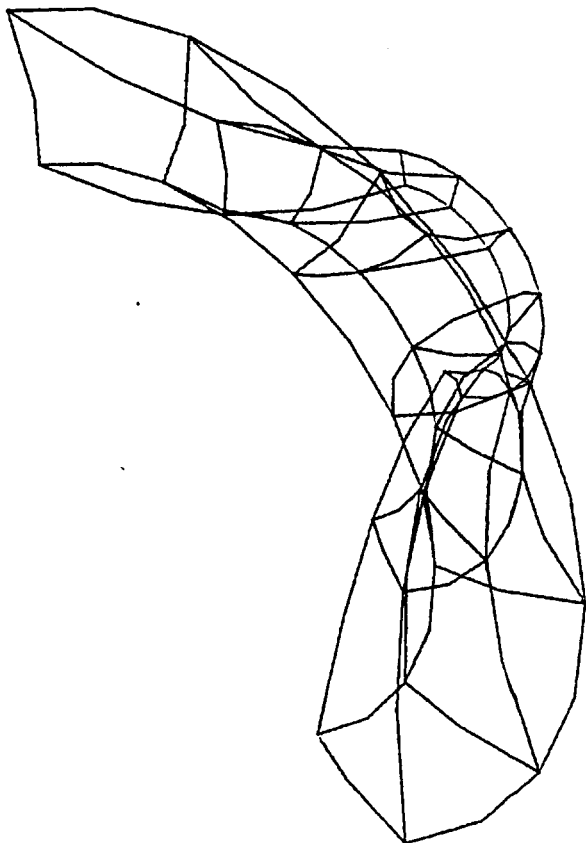


Fig 5.9 Simple 3-D Hexahedral Mesh



Rotation about a skew axis

Fig 5.10

a scaled mesh. The second group consists of all mid point nodes that lie on lines parallel to the Y line in a scaled mesh while the third group consist of all the mid point nodes lying on lines parallel to the Z line in a scaled mesh. For instance in Fig 5.9 the following nodes are in the first group

1,2,3,4,5,9,10,11,12,13, ,17,18,19,20,21
31,32,33,34,35, ,39,40 etc.

By considering the pattern of the node numbers on the first plane ABCD and the next plane at the boundary of the first set of elements in our example, there is a fixed difference between the number of adjacent nodes from one plane to the next. This number is the sum of the total number of nodes on plane ABCD and the number of mid point nodes between the two planes. To simplify the derivation define six quantities:

$NXM = NXL - 1$	The number of mid point nodes on the X line
$NYM = NYL - 1$	The number of mid point nodes on the y line
$NZM = NZL - 1$	The number of mid point nodes on the z line
$NXP = 2 \cdot NXL + 1$	The number of nodes on the X line
$NYP = 2 \cdot NYL + 1$	The number of nodes on the Y line
$NZP = 2 \cdot NZL + 1$	The number of nodes on the Z line

The number of nodes on plane ABCD (NPP) is given by the following equation

$$NPP = NXP(NYL + 1) + NXM \cdot NYL \quad (5.17)$$

The number of nodes on the mid point plane adjacent to ABCD is

given by

$$MP = (XL + 1)(NYL + 1) \quad (5.18)$$

Adding the last two equations and simplifying we obtain

$$\begin{aligned} NDIFFZ &= NYL(4 \cdot NXL + 3) + 3 \cdot NXL + 2 \\ &= NPP + MP \end{aligned} \quad (5.19)$$

Similarly there is a fixed difference between the number of the nodes on line AB and the adjacent ones on line MN and this number is:

$$NDIFFY = NXP + NXM \quad (5.20)$$

It is obvious from the numbering pattern that NDIFFZ is the difference in node numbering when we go from one plane perpendicular to the Z line to a second plane two Z points away. At node 1, $IZ = 1$, at node 22, $IZ = 2$, at node 31, $IZ = 3$, thus our first group of nodes that are numbered are in unit increment of IX but double increments of IY and IZ. The equation that describes the first group of nodes is as follows:

$$I = NDIFFZ (IZ - 1)/2 + NDIFFY (IY - 1)/2 + IX \quad (5.21)$$

when $IZ = 1, 3, 5 \dots \dots \dots 2NZL-1$

$IY = 1, 3, 5 \dots \dots \dots 2NYL-1$

$IX = 1, 2, 3 \dots \dots \dots 2NXL+1$

For the set of nodal points whose numbering is produced by the above equation the scaled coordinates can be obtained using an equation similar to equation 5.16, the only difference being the

definition of R_i

$$U_i = 2R_{ui} - 1$$

$$\begin{aligned} \text{where } R_{xi} &= (IX-1)/(NXP-1) \\ &= (IX-1)/2NXL \end{aligned}$$

The ratios for the y and z directions can be similarly described.

For the second and third group of nodes only the equation of node numbering and scaled coordinates are presented

Group 2

$$I = NDIFZ \cdot (IZ-1)/2 + (3NXL+2) IY + IX - NXM \quad (5.22)$$

$$\begin{aligned} \text{where } IZ &= 1, 3, 5 \dots 2NZL-1 \\ IY &= 1, 2, 3 \dots NYP \\ IX &= 1, 2, 3 \dots NXM \end{aligned}$$

$$\begin{aligned} \xi_i &= [2(IX-1)/NXL] - 1.0 \\ \eta_i &= [(2* IY-1)/NYL] - 1.0 \\ \zeta_i &= [(IZ-1)/NZL] - 1.0 \end{aligned} \quad (5.23)$$

Group 3

$$I = IDZ(IZ) + IDZ2(IX-1) + NXM(IY-1) + IX \quad (5.24)$$

$$\begin{aligned} \text{where } IZ &= 1, 2, \dots NZL \\ IY &= 1, 2, \dots NYM \\ IX &= 1, 2, \dots NXM \end{aligned}$$

$$\begin{aligned} IDZ &= NXP*NYM + NXM*NYL \\ IDZ2 &= NXM*NYM \end{aligned}$$

$$\begin{aligned} \xi_i &= [2(IX-1)/NYL] - 1 \\ \eta_i &= [2(IY-1)/NYL] - 1 \\ \zeta_i &= [(2* IZ-1)/NZL] - 1 \end{aligned} \quad (5.25)$$

By using isoparametric mapping, the use of twenty-point building blocks for the production of eight-node brick elements was possible. The twenty point input building block can fit nearly all curves encountered in engineering design, but the limitation of using isoparametric mapping is that certain curves, for example a circle, cannot be modelled exactly. In the majority of cases the errors produced will be negligible but for small solids of revolution, a simple technique can be used to generate a planar mesh in a plane and revolve this mesh round an axis to produce the required shape without errors in the nodal positions. In the previous chapter this was achieved in a limited form by producing a mesh in the X-Y plane and revolving the mesh about the Y axis, by the use of a two dimensional rotation matrix.

5.2.3 Matrix Transformations

A transformation is normally expressed as a single entity, the transformation matrix, of which there are three main primitive types:

- a. Scaling
- b. Translation
- and c. Rotation

Complex transformations can be expressed as a sequence of primitive transformations, or a concatenated expression yielding a single transformation matrix that has the same effect as the sequence of the primitives.

5.2.3.1 Scaling

A scaling transformation is used to scale dimensions in any of the three directions of a point X, Y, Z to a new point X', Y', Z' . Such a transformation is given by the following equation

$$[X', Y', Z'] = [x, y, z, 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

5.2.3.2 Translation

The transformation which translates a point x, y, z to a new position X', Y', Z' is

$$[X' \ Y' \ Z'] = [\bar{x} \ y \ z \ 1] [T]$$

where $[T]$ is the translation matrix

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ T_x & T_y & T_z & 1 \end{bmatrix}$$

5.2.3.3 Rotation

Three dimensional rotation transformations have more complexity than two-dimensional rotation transformations. We must define a three dimensional axis in space about which to rotate, and if this axis is not aligned with any of the coordinate axis, we must concatenate five transformations.

The first translates any fixed point on the line to the origin.

The second performs the appropriate rotations about the X and Y axes of this coordinate system so that the unit vector of the direction cosines is mapped into the unit vector along the Z axis. The third transformation performs the desired rotation θ about the Z axis of the new coordinate system. The last two transformations are the inverse of the second and first respectively and they are performed to return the object to its original coordinate system. The coordinate systems created by steps 1 and 2 are used only as intermediate steps in the process. The complete transformation representing the five steps is

$$T R_1 R_2 R_\theta R_2^{-1} R_1^{-1} T^{-1} \quad (5.26)$$

Let a, b, c be the direction cosines of the axis of rotation
Let $v = (b^2 + c^2)^{\frac{1}{2}}$, and (x, y, z) the coordinates of a point through the axis of rotation.

The matrices in equation 5.26 are:

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -x & -y & -z & 1 \end{bmatrix} \quad (5.27)$$

$$R_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c/v & b/v & 0 \\ 0 & -b/v & c/v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.28)$$

$$R_2 = \begin{bmatrix} v & 0 & a & 0 \\ 0 & 1 & 0 & 0 \\ -a & 0 & v & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.29)$$

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.30)$$

A subroutine "TRANSM" written on the basis of the above formulation is used in rotating any point in space about any specified axis. But this is only part of the whole rotation program because the production of a twenty-node element from an isoparametric eight node surface element requires a sorting process where all the corner nodes are rotated twice: once for the generation of the mid point nodes on the edges and the second time to generate the other corner nodes. For the first rotation, half the required angle is used. The last two building blocks MACROBLOCK 3 and MACROBLOCK 4 use this rotation technique in generating 8-NODE and 20-NODE element around solids of revolution. For solids of revolution with curved sections, the isoparametric mapping technique is first used in meshing the section, before rotating the mesh about the given axis, Fig 5.10.

5.3 JOINING BUILDING BLOCKS

The production of a unique mesh by joining one or two surfaces of a building block to that of an existing model is a much more complicated affair when compared to joining one

quadrilateral to a planar model. For quadrilaterals the numbers of the nodes on the new quadrilateral are specified manually as shown in the last chapter. To do the same for a three-dimensional mesh is a very imperspicuous task, for there is no easy means for the operator to know the temporary numbers of the nodes on the surface or surfaces of the new building block that will be made common to an existing structure. For instance only for a two by two mesh on the surface ABDC of Fig 5.9, there are twenty-one nodal points, and this surface is the easiest of all six surfaces for node number specification purposes, since the node number starts from one and ends in twenty-one in increments of unity. To specify these node numbers manually is a highly repetitive process and the aim of CAD is to transfer such a process to the realm of machine functions. It was expedient for the user to specify a surface or surfaces on the new building block that should be joined to the existing model, and the computer to sort out the temporary node numbers on the specified surfaces and recognise these nodes as common nodes.

Thus it was necessary to define the surfaces of the building block with numbers. In any building block there are six surfaces and they are thus defined:

- FACE I - The value of the counter IX for brick elements is 1
- FACE 2 - The value of the counter IX or IR is the total number of node on the X line or R line
NX

FACE 3 - The counter IY is 1
FACE 4 - IY = NY
FACE 5 - The counter IZ = 1
FACE 6 - IZ = NZ

By fixing the values of the counters in equations 5.15a or 5.21, 5.22 and 5.24 the nodal points numbers on any of the six surface can be computed. A subroutine 'COMSOT' was written to sort out all the common nodes on any specified surface, and mask these nodes from being renumbered.

For displaying and plotting purposes, masking the common nodes to stop them from being numbered again is not of much importance. The common nodes must have the same coordinates as those on the existing model, to produce a unique mesh. A second subroutine 'FNDCOM' was written to inspect the masked nodal points on the new building block and compare them to those on the existing model, and to regularize any anomalies, by giving the coordinates of the existing nodes to the new nodes on the common face.

All the six surfaces of the new building block can be joined to an existing model by specifying all six surfaces as common. The routine is written in such a way that numbers greater than six or less than one are ignored and such numbers, usually zero is used in ending the specification of common faces. The disadvantage of specifying face numbers instead of node numbers is that the user is not free to join part of a

surface to an existing model, unlike that of two-dimensional joining where part of a line can be joined to an existing figure. The advantages of specifying only the surface number for the program to extract and normalize the nodes involved, and the time saved by the user heavily outweigh the disadvantage. The program is capable of reporting to the user if there are no nodes on the existing structure, close to the surface specified common. When the joining of a larger subdivision is more desirable to a smaller existing part of a structure, the former must be further subdivided in such a way as to make the new common surface equal to, or smaller in size than the existing part. This might seem complicated but in practice it is quite simple. Figures 5.11, 5.12 and 5.13 show a gradual build up of an engineering component.

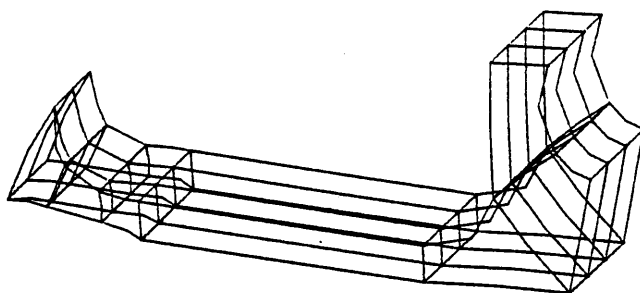


Fig 5.11 First set of macroblocks

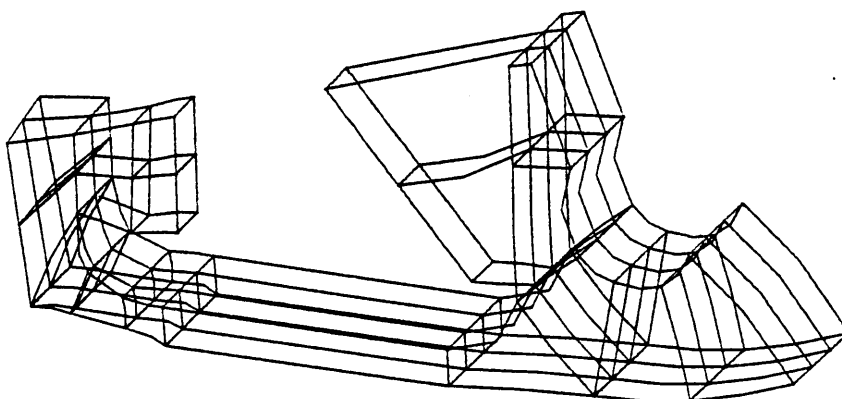


Fig 5.12 Six more macroblock added

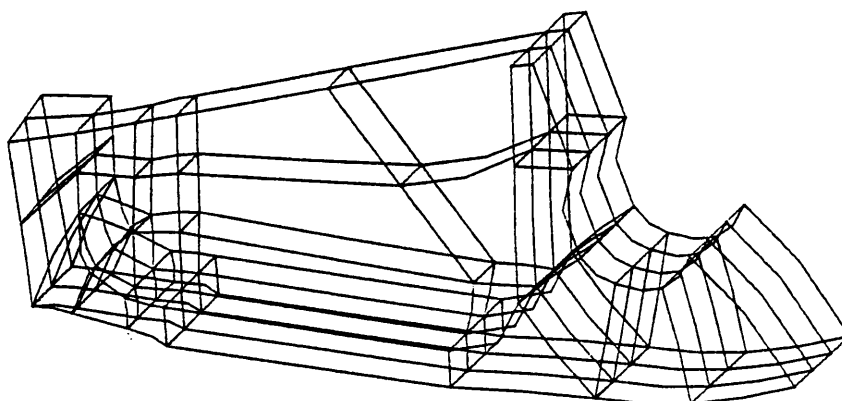


Fig 5.13 Mesh approaching final shape

C H A P T E R 6

DATA HANDLING FOR GRAPHICS AND 3-D MESH

Earlier in Chapter 3 a general account of data storage and file handling is briefly outlined as part of the general system software. Also in the fourth chapter a section deals with mesh data storage and file handling. The current chapter is in a lot of ways complementary to the two mentioned above, but with reference to three dimensional data. For every two dimensional point digitised on the table the subroutine CAMRTB is called transparent to the user by the digitising overlay or any other overlay that required coordinates of points from the digitising table. The function of CAMRTB is to decode the coordinates of the point and send them to the calling program in tenths of a millimetre. These points can either be displayed, stored or plotted. For two dimensional graphics, the coordinates are both stored and displayed by calling the subroutines SETR (see chapter 3 page 42) and PLOTSC respectively. The input of three dimensional data cannot be displayed like that of two dimensional data, as it is stated in the last chapter that two planar points are required to produce a three dimensional point. Thus the basic digitising overlay for the system is not suitable for three dimensional graphics input. DIGOV has been successfully modified for three dimensional data input by Thompson[23]. In SAMG, the functions of a three dimensional digitising overlay are performed by the overlay 3D MESH. The shortcomings of this overlay are:

- a. Although the position of the cursor is echoed on the VDU successive points digitised are neither

stored in the workspace nor echoed on the screen.

- b. The figure being digitised is not seen until all the points for constructing the figure are supplied, leaving the user in a dark alley until all the points necessary are supplied.

As a result of the last difficulty the overlay is supplied, with several editing points where the user can correct an input mistake.

Once the input is right and the data of interest (mesh coordinates) is generated, the need to check and verify the data arises, (CAD).

6.1 DISPLAYING AND PLOTTING 3-D GRAPHICS DATA

3-D graphics data can be displayed on the VDU without being stored in the workspace, but for plotting the data must be stored in the workspace or in a separate file on disk, since it is hardly desirable to plot what we really don't want to keep. It should be pointed out that it is not impossible to plot without storage of the data, this can be achieved by writing stand-alone programs and calling the necessary segment of the plotter driving routines.

In the second chapter it was stated that the final display unit would be the Tektronics 4014. The previous VDU, the TEKTRONICS 611 had a vector generator that required the

first and last points of the vector supplied. Unlike the 611, the 4014 had an internal microprocessor that would store the last set of coordinates and requires only a new point to be the second point for a vector, unless it is coded to produce an invisible vector. As discussed earlier, every three dimensional point can be stored in the workspace or a corresponding resident common buffer in seven sixteen bit words of I, X, Y and Z. I, the pen or beam code occupying the first word of the record, and the X, Y and Z occupying two successive words respectively as long integers in units of tenths of a millimetre.

To display or plot directly from the data stored in the workspace two sets of driver routines have to be written, one for the VDU and the other for the plotter. It would have been desirable to use the same set of routines to drive both the plotter and the VDU, but this was made impossible by the fact that the data intake format acceptable for the plotter was different from that of the TEKTRONICS 4014. One of the major problems encountered in the development of these routines was the basic philosophy of the modularity of the CAD package, where replacement modules are to be written in such a way as to minimise and if possible eliminate changes in other modules that use the module being replaced. The display routines were an indispensable part of the overall CAD package and the new display routines have been written for total compatibility. The plotting routines have also been written to be nearer the display routines in function than the previous set of plotting routines. The design of the routines is such that one routine can send a

drawing to the plotter or the VDU by either calling PLOTAØ or PLOTSC respectively.

Three main methods are available for the visual representation of three dimensional data for displaying or plotting purposes:

- a. Isometric projection
- b. Orthographic projection
- c. Orthogonal views

Isometric projection

The isometric projection overlay whose routine was originally written by McClintock [24], has been modified by the author to produce isometric drawings on the VDU and the plotter. Thus instead of writing two separate routines, one for the VDU and the other for the flat bed plotter, a single routine has been written based on the previous routine since the basic theory of isometric transformation remains unchanged. By defining the origin of the isometric projection as $XØ$, $YØ$ the following equations are used to produce the isometric data

$$\begin{aligned} x_i &= XØ + (X(I) - Y(I)) \cos 30 \\ y_i &= YØ + ((X(I) + Y(I)/2) + Z(I)) \end{aligned} \quad (6.1)$$

where $X(I)$ is the X coordinate value of point I in the work space

$Y(I)$ the Y coordinate value

$Z(I)$ the Z coordinate value

X(I), Y(I) and Z(I) are extracted from the workspace file on DKØ, the system disk. The pen code for every record is also read and used by the plotting routines. The user doesn't have to concern himself with the choice of plotting routine since "DISPLAY ISOMETRIC" and "PLOT ISOMETRIC" are menu commands for sending the isometric drawing to the VDU and plotter respectively.

Orthographic projection

For complicated three dimensional wire frame drawings as in the case of mesh generation, the visualization of these figures using isometric projections can make things difficult since the angle of projection is fixed. The user might in most cases understand the drawing better if he can look at it from different angles. When an object is rotated, the viewer could be confused if the new view is not clearly represented, as might be the case in wire frame drawings where it is difficult to distinguish between lines representing the front and the rear. To aid the CAD system user to interpret two dimensional projections of three dimensional objects, there are several visual aids available.

Perspective transformation is one of the most valuable techniques for visualizing wire frame modules. It is usually employed in the display of three-dimensional data, but can also be used with two-dimensional data to represent a plane lying in three dimensional space. The mathematical representation of perspective transformation can easily be derived since it is based on elementary optics, and is therefore most suitable for

computer manipulation.

Consider Fig 6.1 where $\bar{x}$, $\bar{y}$, $\bar{z}$ represent the axes corresponding to the spatial cartesian coordinate. For an observer with the eye positioned at E (x_e, y_e, z_e), and the directions of sight at angles ($\alpha \beta \gamma$) to the major axes, the point P (x_p, y_p, z_p) in space is mapped onto a projection plane, at a distance D from the observer, by the following relations:

$$X = \frac{(S_x - R_x) \cos \gamma (S_z - R_z) \cos \alpha}{\sin \beta} \quad (6.2)$$

$$Y = (S_y - R_y) / \sin \beta$$

where $[X, Y]$ are the coordinates of the projected position of the point P

$\bar{S} [S_x, S_y, S_z]$ is the point of intersection of the line of sight on the projection plane

$\bar{R} [R_x, R_y, R_z]$ the foot of the normal from the observers eye to the plane.

Appendix C contains a detailed mathematical derivation of the above equation. Perspective transformation is extensively used throughout the system and is regarded as the standard way of viewing graphical output. An advantage of perspective is that, it is defined by only a few parameters, related to the position of the observer. Thus a new perspective view can be generated simply by redefining these parameters, leaving the object unchanged.

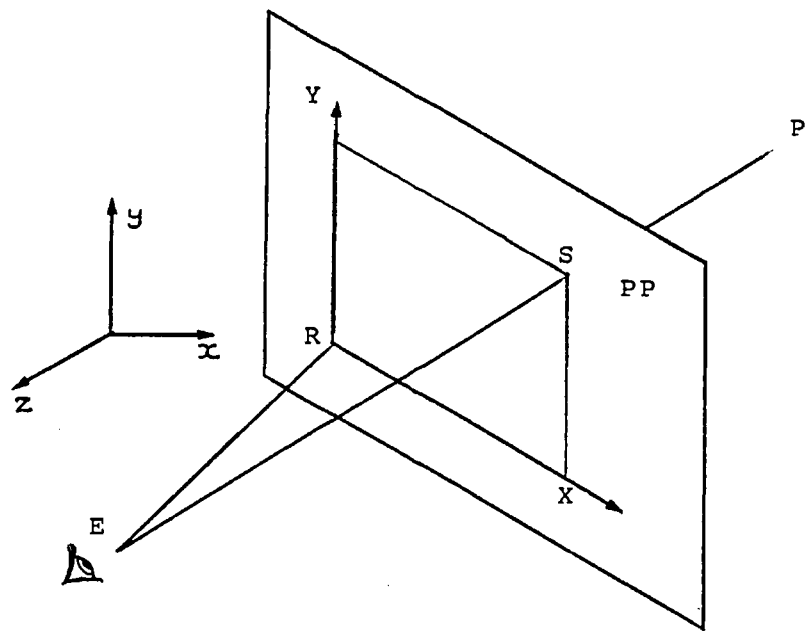


Fig 6.1
Projection on plane PP

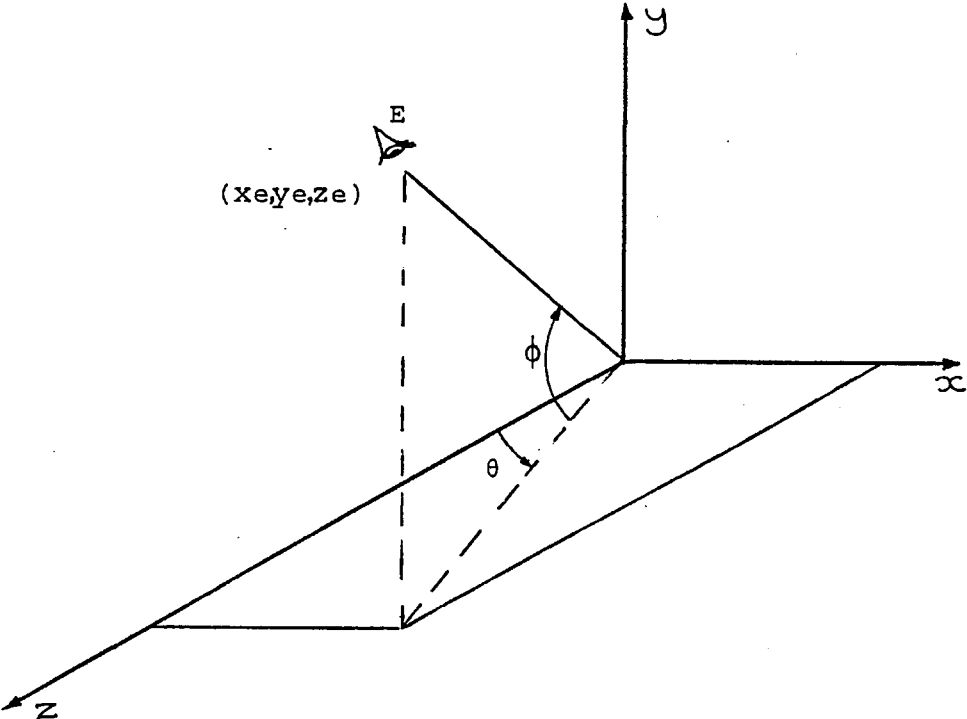


Fig 6.2
Eye position in terms of pan and tilt angles

Once the data concerning location and direction of the observer's eye is defined, a program sets up and positions a viewing plane, at a given distance from the observer, on which the perspective view of the object can be projected. The ability to easily vary the perspective parameters allows a very convenient solution to a joystick controlled display.

The joystick functions allow the operation of a joystick to be emulated on a patch within the menu region by pressing the button over the 'JOY-STICK' menu square, the lower left hand corner of the menu region can be used in controlling two functions rotation and zoom, by pressing the button again over the patch. Rotation is performed about the Y axis and an axis on the X-Z plane determined by the pan and tilt angles respectively. (θ and ϕ in Fig 6.2).

The eye position is defined by $[E, \theta, \phi]$ where

E is the distance from centre of attention

θ is the pan angle

ϕ is the tilt angle

The coordinates of the eye (x_e, y_e, z_e) with respect to the centre of attention are given by the following equations.

$$\begin{aligned} x_e &= E \cos \phi \sin \theta \\ y_e &= E \sin \phi \\ z_e &= E \cos \phi \cos \theta \end{aligned} \tag{6.3}$$

and the direction cosines

$$\begin{aligned}
 \cos \alpha &= x_e/E \\
 \cos \beta &= y_e/E \\
 \cos \gamma &= z_e/E
 \end{aligned}
 \tag{6.4}$$

Conversely

$$\begin{aligned}
 E &= (x_e^2 + y_e^2 + z_e^2)^{\frac{1}{2}} \\
 \theta &= \tan^{-1} (x_e/z_e) \\
 \phi &= \sin^{-1} (y_e/E)
 \end{aligned}$$

The observer is always assumed to be facing the origin of the coordinate system. For cases where the object position is out of range, it can be repositioned by using a centralizing function, which automatically places the object with the centre of volume at the origin.

It is easier to create the movement of an object by varying the observer viewpoint than the actual object position. With the centre of attention inside the object, the object is always visible on the screen. The coordinates of the object in the data file (RA 1 known as the three dimensional workspace FILE 1) remain unchanged, only the perspective parameters are updated.

Rotation is obtained by varying the two angles ϕ and θ in Fig 6.2, and the relationship between these angles and the direction cosines $\cos \alpha$, $\cos \beta$ and $\cos \gamma$ are given by

$$\begin{aligned}
 \theta &= \tan^{-1} \frac{\cos \alpha}{\cos \gamma} \\
 \phi &= \tan^{-1} \frac{\cos \beta}{\sqrt{\cos^2 \alpha + \cos^2 \gamma}}
 \end{aligned}$$

The zoom factor being determined by the ratio between the distance of the plane of projection and the observer distance (D/E).

Armed with a joystick routine which can easily change the perspective transformation parameters, we are capable of viewing our object from virtually any angle we choose. All the display and plot routines are written to request for the perspective transformation parameters before display if and only if we want to display the data in FILE 1. On transformation the coordinate of the points as displayed on the screen, are stored in FILE 2. Since the perspective transformation is always requested before display, and the time involved in the updating of the parameters is relatively insignificant, the resultant increase in display time is minimal. For the plotting there is no change at all in the plotting time since it is controlled by the speed of the plotter which is very slow compared to the speed of the computer.

Orthogonal views

This facility projects the data in the three dimensional workspace FILE 1, on to three major planes which are then unfolded for presentation on the screen. It is sometimes called direct viewing and the process is controlled by an overlay called "DVIEW". Each view is generated by taking the coordinates' terms in FILE 1 corresponding to the plane being considered. Thus from an [I, X, Y, Z] record only X, Y are used for the X - Y plane. The figure thus produced will be identical to a

FILE 3 data display as it were presented or created along with FILE 1. Alternatively an overlay square is provided for generating FILE 3 from FILE 1 and the transformation from FILE 1 to FILE 3 is also carried out by the same overlay "DVIEW". To produce the Z-X and Z - Y planes the overlay use the coordinates of the Z datum (ZX, ZY), the point at which all the three orthogonal planes meet.

To plot or display the Z-Y plane the coordinates of any point in the [1, X, Y, Z] record will be [x_{zy}, y_{zy}]

$$\text{where } x_{zy} = ZX + Z$$

$$y_{zy} = Y$$

for x - z plane

$$x_{xz} = X$$

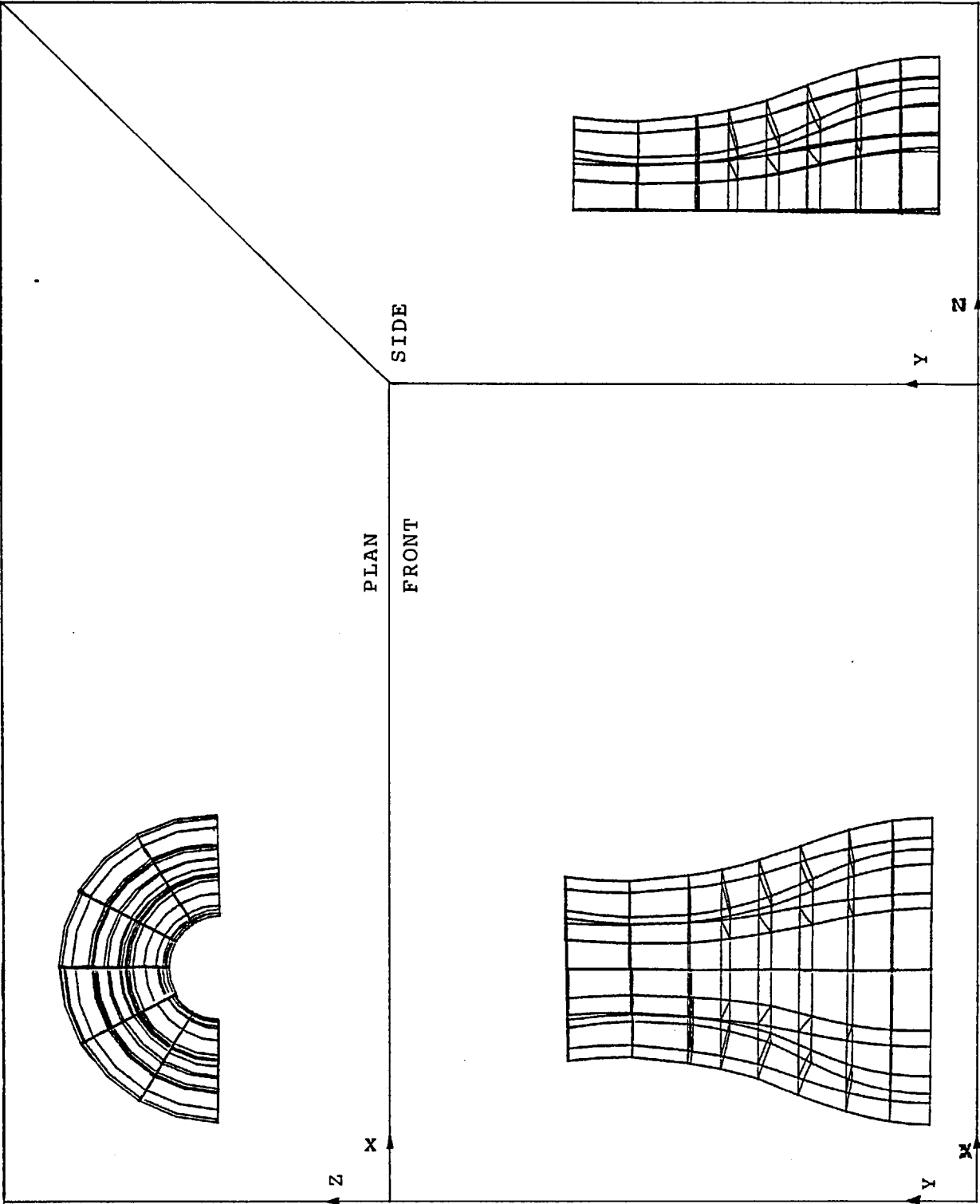
$$\text{and } y_{xz} = ZY + Z$$

The overlay responsible for this type of display or plotting does not need any perspective routine or transformation only the data stored in FILE 1 and the appropriate shifts to place each view within its own quadrant. Fig 6.3 shows three orthogonal views of a mesh.

6.1.1 Plotter Character Generation

Unlike the TEKTRONICS 4014 (the main visual display unit) which has a hardware character generating set, the plotter needed a software character generating routine. Without such

Fig 6.3 Orthogonal views of mesh generated for a cooling tower



a routine, labelling of drawings and generating the node numbers of mesh on the plotter were impossible tasks. Understanding the data intake format to produce a vector generating routine was a major task and since every letter, alphabetic or numerical can be divided into series of vectors, the character generation program was in sight. The development of such a program was not as easy as it seemed at first sight and at the time of writing only upper case character have been supplied to the bitmap of the character generating set.

6.2 3-D MESH DATA

The displaying and plotting of 3-D mesh data are in accordance with the 3-D graphical data handling described in the previous section, as long as the data is sent to the responsible routine in the proper sequence. To explain the above sentence, let us consider the general structure of the 3-D mesh data. First, the basic building block is meshed into smaller blocks or elements and the coordinates of each node is stored in an unnamed common block within the mesh generating overlay. The node numbers at this stage are temporary and are not modified until it is necessary to create a coordinate data file. The next action of the mesh generating subroutine is to inquire from the user if there is a common face (ie if any of the faces of the block that has just been mesh^{ed} is to be joined to an existing figure in the work space). If there is any common face, a subroutine 'FNDCOM' is called, whose function is to search the workspace and trap all the coordinates of the nodes on the existing figure, near our new building block. The nodes

on the common faces are then compared to those of the existing figure and if there is any difference between the coordinates of any two nodes that should be coincident, a synchronization process is carried out by changing the coordinates of the nodes in the new building block to those of the existing nodes.

'FNDCOM' the common nodes finding routine is very important to the strategy used in the overall mesh generation module.

Should the user supply the wrong common face number, the routine 'FNDCOM' is capable of detecting this error and will flash the following message on the VDU screen 'NO NEAR NODE'.

DISPLAYING MESH

In order to display an eight-node brick element, a total of sixteen points are supplied to the display routine. For twenty node isoparametric elements, a total of twenty eight. For brevity only the eight node brick element is considered, and for every element the node numbers supplied to the display routine are calculated as follows:

```

Node 1  =  IX + (IY-1)* NXP + (IZ-1)* NXP*NYP
Node 2  =  Node 1 + 1
Node 3  =  Node 1 + NXP+1
Node 4  =  Node 3 - 1
Node 5  =  Node 1
Node 6  =  Node 1 + NPP
Node 7  =  Node 2 + NPP
Node 8  =  Node 3 + NPP
Node 9  =  Node 4 + NPP
Node 10 =  Node 6
Node 11 =  Node 4

```

```

Node 12 = Node 9
Node 13 = Node 3
Node 14 = Node 8
Node 15 = Node 2
Node 16 = Node 7

```

where IX, IY and IZ are counters in the ξ , η and ζ direction of the scaled macroblock respectively.

NXP, NYP are the total number of nodes in the ξ and η direction respectively and

NPP is the total number of nodes in the first plane..

$$NPP = NXP * NYP$$

After defining these nodes a four line loop generates the figure using the sixteen points in the order from 1 to 16, making sure that the pen code is 1 (ie, pen up or invisible vector) when the node number is 1, 11, 13 or 15.

The displaying sequence is the same as that of the storage sequence in the work space. Thus for every element there are eight more points stored than the number of nodes on the element. It seems an inefficient way of storing the mesh data. This can be improved by writing a routine that will accept only the nodes on the element to generate the figure by calling a macro generating routine. So far this improvement has not been carried out, since at the development stage the bulk of the programs written were primarily for test purposes. Such an improvement will cut down the size of store required by exactly half for eight-node elements, and a 40% savings in store for twenty-node elements.

COORDINATE DATA FILE

The coordinate data generated are stored in permanent files very much like those of 2-D mesh, and for standardisation purposes, the same kind of name is given to all coordinate data files in two or three dimensional mesh. Thus a call to STORE (IFILE) described in Chapter 4 will create a permanent file on the system disk for our data. Although the basic format of the data file is the same for both two dimensions and three dimensions, the requirements of the 3-D coordinate data file generating program is quite different.

In 2-D coordinate data files, the common nodes supplied by the user on the temporary numbering are simply masked out and the remaining nodes numbered directly from the first node-number supplied. This is possible and acceptable because the creation of the element connection files is independent of the coordinate data file. Since these two inter-related sets of files are created by different menu commands the supply of parameters such as number of element or nodes in each direction of the scaled model is repeated from one set of files to the other. For 3-D mesh data, the mesh monitor overlay '3D MESH' loads appropriate overlay and segment in such a way as to create the topology data immediately after the coordinate data. Thus we cannot simply mask out the nodes on the common face and create a data file without these common nodes. Instead the nodes on the common faces of the new building block are masked in such a way that the corresponding node numbers in the previously created coordinate data files are recovered

and written in the new coordinate data file. Only nodes that are not common to the existing figure, are numbered starting from the first node number. Before the coordinate data files of two adjacent building blocks in a figure are joined, the common nodes on the common surface appear in each data file. Two subroutines developed by the author 'COMSOT' and 'XCOR' are called for sorting out the common nodes for masking, and recovering the node number of the common node in the previous data files respectively. Effectively if there are 81 nodal points in the mesh pattern the temporary numbering will start from 1 to 81, after supplying the common faces and the first node number, there will still be 81 nodal point numbers, but some of the number will be less than the first node number, and the last node number will be less than the first added unto eighty. If IFT is the first node number and ILT is the last node number and any of the faces of the building block is declared common, then

$$ILT < IFT + NNP - 1$$

where NNP is the total number of nodal points in the mesh.

TOPOLOGY DATA FILE

Once the creation of the coordinate data file is complete, the overlay for creating the topology data file is loaded while all the parameters common to the coordinate file and the topology file are stored in a resident common block. One overlay is used for both twenty-node and eight-node elements thus the user is required to guide the system when a request

is made on the issue of the number of nodes per element.

Earlier in Chapter 4, it was stated that the coordinate data file can be used by nearly all known finite element programs by supplying the necessary code to every line of the data file. This statement which is true for 2-D data file is also true for 3-D coordinate data. But in the case of the topology data, whereas there are only eight possibilities of listing out the order of the node numbers within each quadrilateral element, there can be as many as $20!$ possibilities for the unrestrained twenty-node element and $8 \times 6!$ possibilities for the restrained eight node element. Thus the creation of a topology data file will in general conform to a particular solution program. In SAMG, topology data files are produced to conform with PAFEC input data. [28]

For an element whose origin node 1 (as shown in Fig 6.4) is defined by IX, IY and IZ (which are the counters in the ξ , η and ζ directions of the scaled model), the topology is defined as the twenty node numbers contained in that element and presented in the order shown in Fig 6.4

The value of each node is calculated as follows:

$$\begin{aligned}
 \text{NODE 1} &= 2 \cdot \text{IX} - 1 + (3\text{NXL} + 2)(\text{IY} - 1) + (\text{NYL}(4\text{NXL} + 3) + 3\text{NXL} + 2) \cdot (\text{IZ} - 1) \\
 \text{NODE 2} &= \text{NODE 1} + 2 \\
 \text{NODE 3} &= \text{NODE 1} + 3\text{NXL} + 2 \\
 \text{NODE 4} &= \text{NODE 3} + 2 \\
 \text{NODE 5} &= \text{NODE 1} + \text{NYL}(4\text{NXL} + 3) + 3\text{NXL} + 2 \\
 \text{NODE 6} &= \text{NODE 5} + 2 \\
 \text{NODE 7} &= \text{NODE 5} + 3\text{NXL} + 2
 \end{aligned}$$

```

NODE 8  =  NODE 7 + 2
NODE 9  =  NODE 1 + 1
NODE 10 =  NODE 9 + 2NXL+1 - IX
NODE 11 =  NODE 10 + 1
NODE 12 =  NODE 3 + 1
NODE 13 =  NODE 1 + 3NXL NYL+2 (NXL+NYL) - (2NXL+1)(IX-1)
                                           - IX + 2

NODE 14 =  NODE 13 + 1
NODE 15 =  NODE 13 + NXL+1
NODE 16 =  NODE 15 + 1
NODE 17 =  NODE 5 + 1
NODE 18 =  NODE 17 + 2NXL + 1 -IX
NODE 19 =  NODE 18 + 1
NODE 20 =  NODE 7 + 1

```

Where NXL is the total number of elements along the ξ line
and NYL is the total number of elements along the η line

Similarly the value of each node is calculated using eight separate equations in describing the topology of eight-node elements.

For six-node wedge elements and fifteen-node isoparametric wedge elements, the topology data is written on disk similar to those of eight-node and twenty-node elements respectively, with some of the nodes repeated Fig 6.5. Repeating any node number in an element will produce a zero length along one of the edges of the element. Since most finite elements programs cannot cope with zero length definition, the topology description of wedge elements might at first sight seem a serious handicap to the system. The zero length problem is solved in the data files joining overlay which will be seen next.

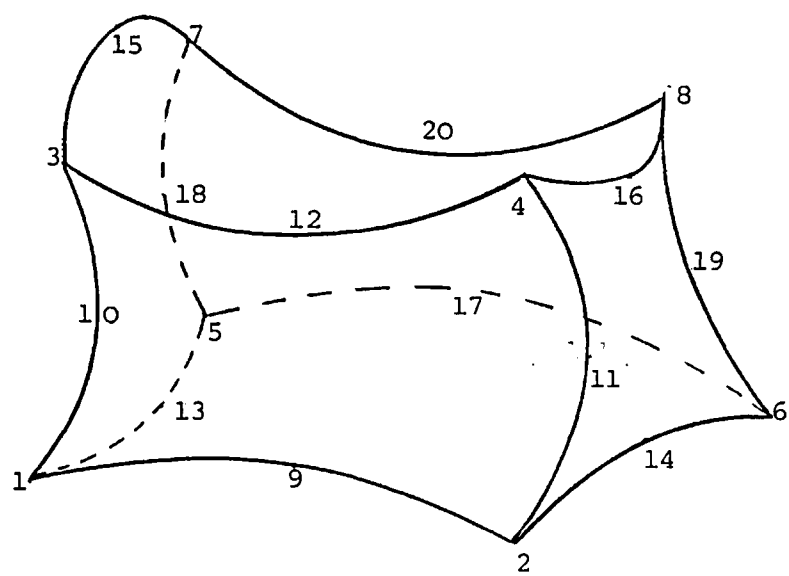


Fig 6.4 20-Node Isoparametric element

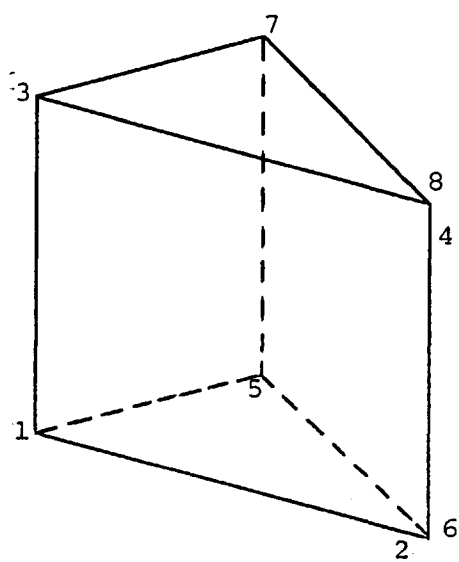


Fig 6.5 6-node wedge element

To overcome the limitation imposed on the user by the topology generating routine, it is desirable to produce, a topology interchange program which can read one topology data file and convert it to be used in another finite elements program. At the time of writing this has not been achieved yet, and it is hoped that this can be achieved at the end of the work.

JOINING DATA FILES

The advantage of creating separate data files for each building block has already been discussed in Chapter 4, for 2-D data files, and these advantages are still applicable for 3-D data. When the user is satisfied with the data in each file or after performing the necessary editing task on each data file, all the files are concatenated into one coordinate data file and one topology data file by the overlay 'JOIN3D'.

On entry to this overlay the user is asked for the kind of file to be joined (coordinate data or topology data). And like the two dimensional file joining, the numbers of the files to be concatenated are supplied by the operators, (using the number entry patch in the menu region) in the order in which they are to be joined. A maximum of twenty files can be specified for concatenation, and if there is an input error, the whole process can be started by specifying the negative sign, 'minus' as a file number. At the end of file number specification, the supply of a zero file number or pressing the digitising pen above the 'ENTER' square on the number entry

patch will advance the program sequence out of the file number input loop. The file numbers entered so far are stored in a named common block.

The creation of one file from several separate files demand the opening of at least two files on the disk for nearly all times of the concatenation process. A dummy file M999.DAT or C999.DAT, depending on whether we are joining coordinates or topology data file, is opened on logical unit 4 on the system disk. Then the files specified for concatenation are opened sequentially on logical unit 7 and their contents transferred to the dummy file. At the end of the transfer operation these specified files are closed before the next is opened. Unlike the 2-D data files, the 3-D coordinates files have repeated node numbers in files of macroblocks that are joined to each other in the digitising process. Thus if we simply transfer the contents of the macroblock mesh data file into the dummy file, some nodes will appear twice or more in the concatenated file. Again for wedge shaped elements there are repeated nodes on the element in the topology data files, and a simple transfer can make the data useless for most finite element programs.

Therefore, the major task of 'JOIN3D' is not a simple concatenation of data files. It is more of a tidying up overlay that joins the files and tidies up all repeated node cases and produces a final data file for each set of data. Once this process is complete, all the smaller data files are deleted and a new number is requested by the system for the concatenated file.

CHAPTER 7

CASE STUDY

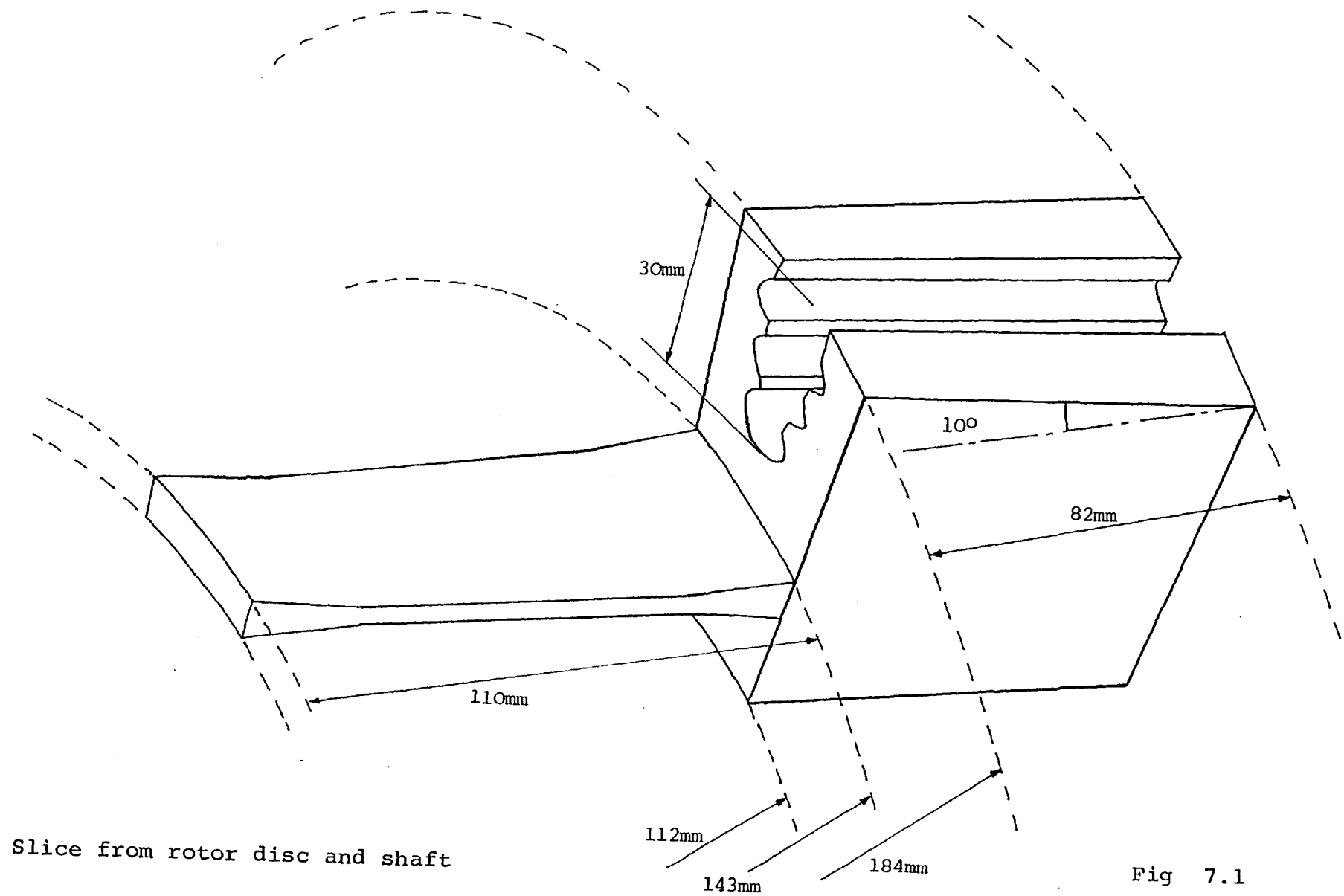
SAMG, a stress analysis mesh generation package has been successfully used in the generation of mesh data for a number of problems. Some of the problems cannot be mentioned in this work due to security reasons. Therefore the case study that follows is by no means the only successful application of the package, but provides a practical example for the use of the system as a preprocessor for a finite elements analysis package FINEL [29]. The mesh coordinates data, mesh connection data, load data and boundary condition data for the following problem were all generated using SAMG within the eight hours of a working day. These data were written onto a magnetic tape and transferred to the CDC* main frame computer for use as input data for the FINEL program.

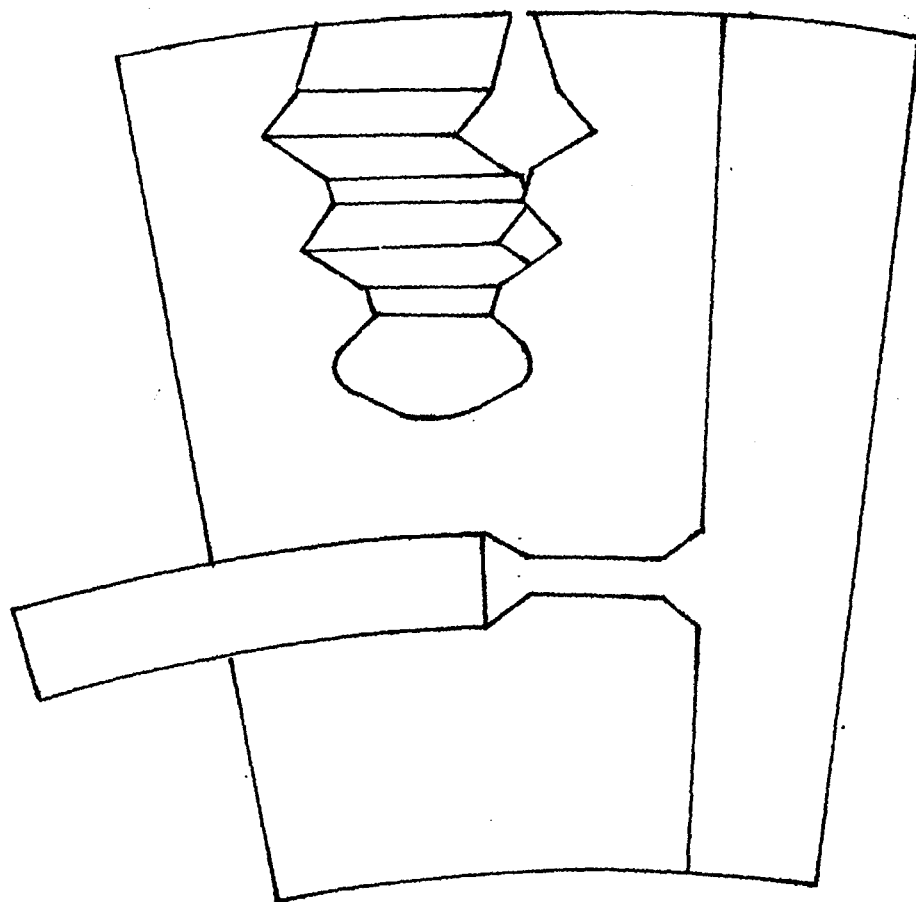
7.1 PROBLEM SPECIFICATION

The problem for the case study was provided by Rolls-Royce Ltd, Bristol as a realistic test case for a complicated three dimensional analysis. Fig 7.1 shows the sketch of a slice from a rotor disc containing a fir tree slot. The slice representing 1/26th of the complete rotor and is cut at an angle of 10° to the axial centre line of the engine.

To generate the mesh, only two drawings of a cross-section in the axial-radial plane and a true view up the slot were supplied, Fig 7.2 and Fig 7.3. Since the fir-tree slot is

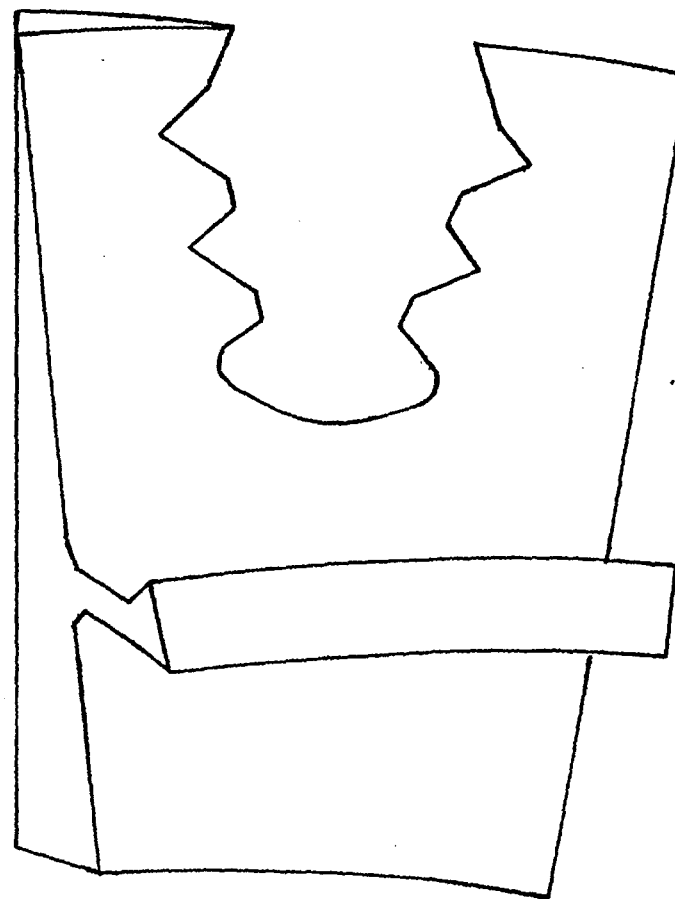
* CDC Control Data Corporation





AXIAL VIEW ON ENGINE

Fig 7.2



VIEW ALONG SLOT 10°

Fig 7.3

also broached at an angle of 10° to the axial centre line of the engine, the axial views on the front and rear faces of the disc were not identical. A proper engineering drawing of at least two orthogonal views of plan and front elevation were required for use in SAMG. Further sketches were provided to specify the dimensions required to produce the drawings of the two necessary views. The time required for the drawing was two and a half hours.

The mesh specification was that it should comprise twenty-node isoparametric elements. Since FINEL was being used as the analysis program, a further constraint of the total number of elements not exceeding five hundred was imposed on the mesh.

7.2 SOLUTION

The whole structure was divided into twenty-two macroblocks, the first nineteen on the disc and the rest on the shaft. The first seven, the twentieth and the last macroblocks were described with twenty input points, while the remaining thirteen were described by eight points per macroblock. The points on each macroblock numbered from one to eight or one to twenty as the case may be were digitised, and the macroblock meshed into a convenient number of twenty-node isoparametric elements as specified. Earlier in Chapter 5, it was stated that the routines for producing the topology data were written to produce the data in a format suitable for the finite element program PAFEC [28]. Therefore it was necessary to write a

simple conversion routine between PAFEC and FINEL to rearrange the coordinates and topology data in a format acceptable to FINEL. The total time required to produce the mesh data was five hours.

7.2.1 Use of FINEL

The following sets of data were required by FINEL for the solution of a stress problem:

1. Mesh Coordinates
2. Mesh Connections
3. Fixed Freedoms
4. Load Points
5. Material Properties
6. Element Type

The first two requirements were fully supplied by SAMG after the topology modification. For the coordinates of the mesh, the node numbers were not required by FINEL since the program assumes an entry of nodes in increasing order starting from unity. Secondly the element numbers in the mesh connection data were not required. Therefore the data generation programs were slightly modified to mask out these unnecessary data by inserting a semi-colon before and after the numbers that are not required. These numbers are still present in the data file for any other finite element program that might require them, and the element plotting routine in SAMG.

A new overlay "FIXFDM" was written to generate the fixed

freedoms or boundary conditions automatically for FINEL, while another overlay for generating the point loads manually "LOADAT" was also written for use with FINEL. Since the loading on the specimen is centrifugal and cannot be easily described as point loads or transformed into equivalent point loads manually, it was necessary to write a third routine to produce the equivalent point loads at the nodes on the surfaces that are loaded centrifugally. Although the last routine has been specifically written for the problem of the case study, it can easily be developed further to generate the loading data for uniform and uniformly changing distributed loads. The loading of the specimen is shown in Fig 7.4b and Table 7.1, and they both show a distribute load changing from one value on the front to another at the rear. Since the surfaces of the tooth that are loaded are plane and not curved, the element surfaces loaded are rectangles. The proportion of the total force acting at the nodes on the element surface given in Fig 7.5d and Fig 7.5e are superimposed to give the equivalent point loads. Using the above formulation and considering the direction cosine of the pressure load on each surface a routine "LOADRR" was produced to generate the load data for FINEL.

7.3 RESULTS AND COMMENTS

To test the package SAMG for mesh generation, the overall mesh of the initial figure (Fig 7.6), was produced on the flatbed plotter from the resulting coordinates of the nodes generated by SAMG. Fig 7.7 and Fig 7.8 show the mesh at various angles to test for any discontinuity, and were produced by the plot

BLADE LOADS FOR DISC MODEL

BLADE CENTRIFUGAL FORCE 250,000 N
BLADE TORQUE 450,000 N mm

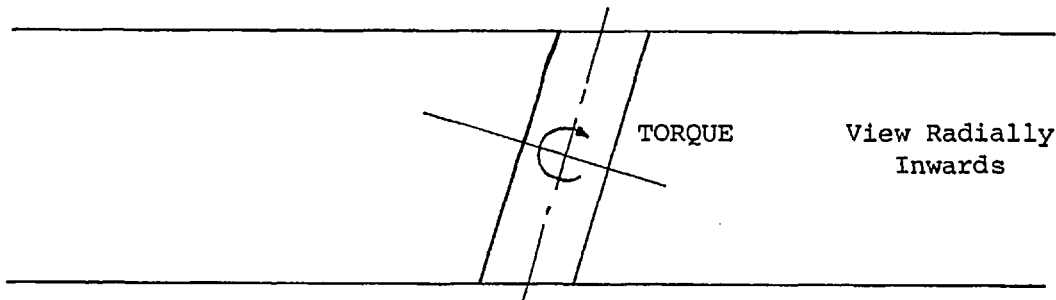


Fig 7.4a

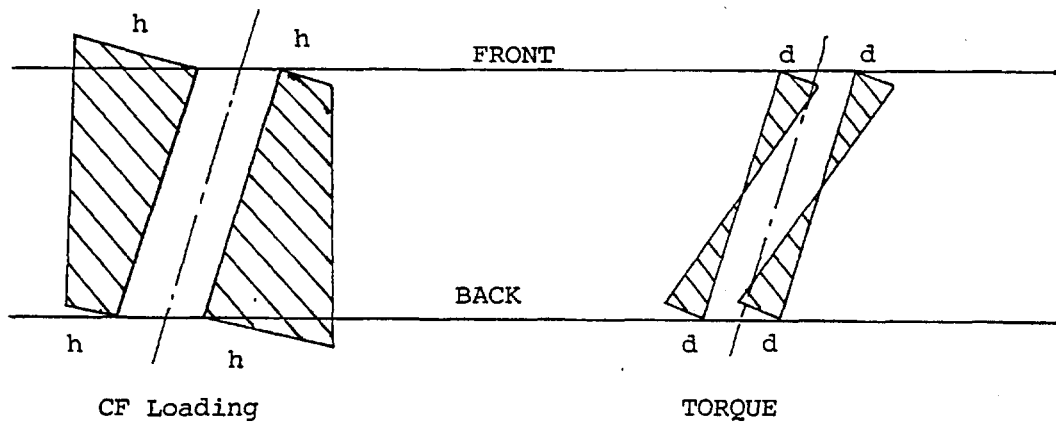


Fig 7.4b

Tooth Normal Loads (N/mm)

		Top tooth	Middle Tooth	Bottom Tooth	
Blade CF	h_1	609.2	630.0	654.6	
	h_2	799.3	778.5	753.9	
Torque		90.9	90.9	90.9	
TOTAL	Front	L	708.4	687.6	663.0
		R	700.1	721.0	745.5
	Rear	L	700.1	721.0	745.5
		R	708.4	687.6	633.0

Table 7.1

EDGE AND SURFACE LOADING FOR QUADRATIC ELEMENTS UNDER DISTRIBUTED LOAD

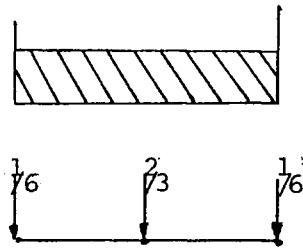


Fig 7.5a

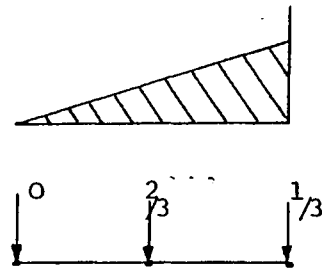


Fig 7.5b

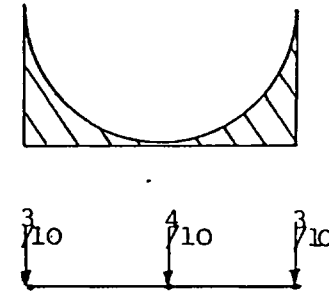


Fig 7.5c

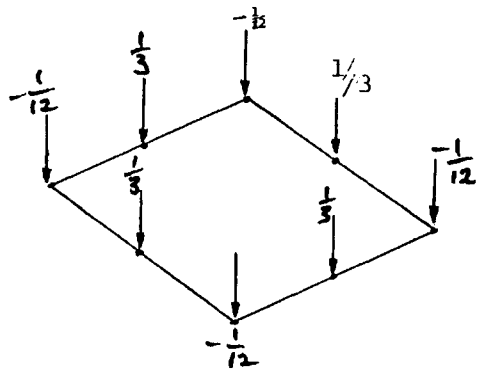


Fig 7.5d

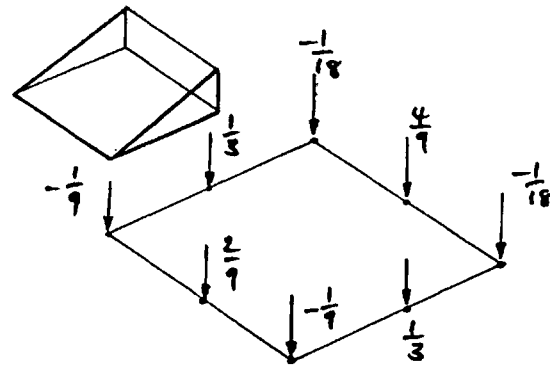


Fig 7.5e

subroutine in FINEL. The results of stress analysis and the mesh data are presented in Appendix C. The occurrence of the peak stresses at the bottom of the root of the fir-tree is in accordance with the results obtained at Rolls-Royce Ltd.

On the whole it has taken two working days in generating the mesh data and solving the stress analysis problem as compared to four weeks spent at Rolls-Royce in producing a specific program to generate the mesh data for the specimen, and use a finite element package. With due respect, it is not easy to produce a program to generate such a complicated mesh, and this will be uneconomical since the program is restricted to just one problem, and can at best be modified for problems of the same class. A more general and flexible package such as SAMG will be an advantage to most stress analysts using finite element methods.

OVERALL MESH FOR CASE STUDY SPECIMEN

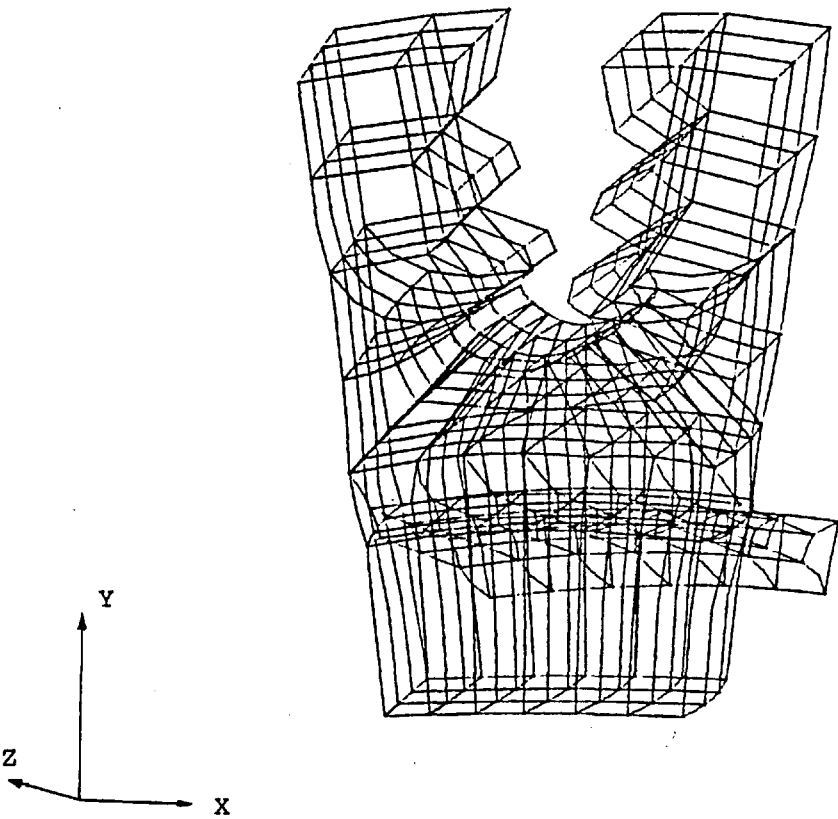


Fig 7.6

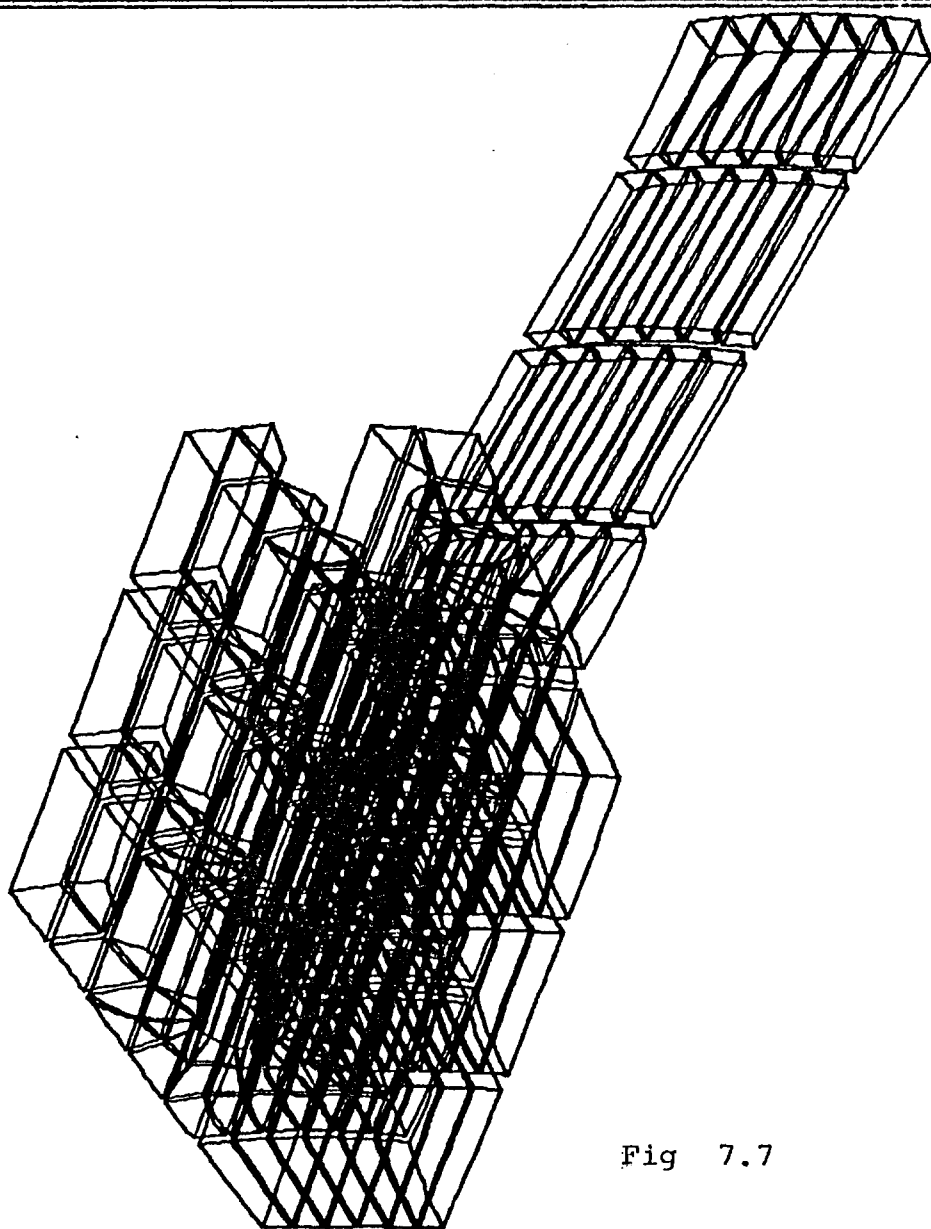


Fig 7.7



Discontinuity test plot

DRAWN BY FINEL
DATE: 81/11/90

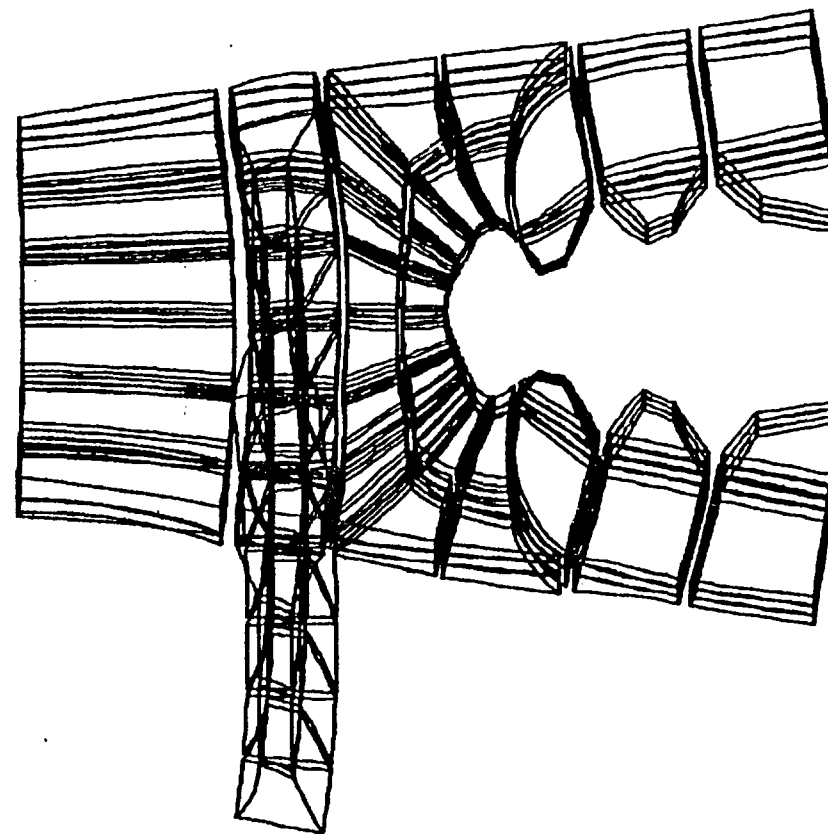
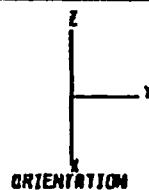


Fig 7.8



Discontinuity test
plot

DRAWN BY FINEL
DATE: 81/12/01

C H A P T E R 8

CONCLUSION AND FUTURE DEVELOPMENT

The work described in this thesis demonstrates the use of computer aided design (CAD) techniques in the generation of complicated three dimensional shapes and subdividing these shapes into meshes for 3D finite element stress analysis. It also demonstrates that shape definition of complicated three dimensional structures which is normally exclusive to main-frame computers, can be handled on a mini computer. The main objective of CAD techniques are to maximise the accuracy of the design work and minimise the time required and the drudgery.

The need to obtain more accurate solutions on technically sensitive areas such as the modern aircraft industry has expanded the scope and use of Finite Element Methods. The wider use of finite element analysis for every day engineering problems has created the need for input-data handling packages (pre-processors) to make the method more attractive and less time consuming. The work described satisfies the above need, and the system thus developed can handle any complicated two and three dimensional shape provided the shape can be drawn.

Although the system is based on a general computer aided design package developed at Imperial College between 1969 and 1973, the mesh generation modules can operate as a satellite to any existing graphics software package with small modification to the display routines. Similarly the mesh generation modules

can also be modified to operate as a stand alone program on a mainframe computer. The overall system described in the thesis has been designed for maximum flexibility and a good degree of generality. The interactive nature of the system gives the user a fair amount of control over the final idealisation. Although the operation of the system is mostly self explanatory and should be easy to master, there should be a short learning period for every user. The learning period should not be used only to learn how to operate the system, but to discover its full potentials and to learn to plan the work in order to exploit these potentials to the fullest. The system depends on a digitising table as the main input peripheral. Should a digitiser not be available then the input routines will need to be rewritten to accept data from either the teletype key-board or some other form of peripheral device. The latter form of input could increase the time for data preparation due to the interactive nature of the programs and a slower response from the keyboard as compared to the digitising table with its menu facility.

The shape definition can be applied to any system since it is neither hardware dependent nor hardware controlled. Once the input coordinates are supplied in the right sequence, the object of interest or the building block will be properly defined before proceeding to the mesh generation. The system will enable the user to generate uniform and non uniform meshes for two dimensional and three dimensional structures. In the development care has been taken not to discard previously existing modules of the parent CAD package. Where necessary,

modification has been done for better operation by the use of more efficient algorithms, and by the introduction of new hardware into the system. Most of the modules for two dimensional mesh generation and the graphics data handling were either changed completely or modified for greater efficiency.

The use of the system as a pre processor of finite element methods can be further improved in the following areas:

- a. Developing a more efficient plotting routine for the mesh data, in order to save memory in workspace
- b. Developing a full data interchange program that will convert the basic data generated to be used by any finite element program
- c. Producing a general automatic loading program for different types of pressure loading, by extension of the loading routine produced for the case study
- d. The introduction of more complex type of building blocks
- e. At present all the graphic input is point based and by going one step further to a line based input or two steps to a surface base input, the operation of the system might be more efficient considering the time required in defining surfaces and objects.

The use of the system as a post processor is only limited to two dimensional analysis and it will be desirable to extend this to three dimensional problems. Any development in this line should preferably be accompanied by the introduction of more powerful hardware such as a colour graphics terminal.

Potentially, some of the algorithm^s developed in sorting out the common nodes of building block surfaces could be further developed to form the bases of a small hidden-line removal program. The same technique of overlaying programs for draughting and mesh generation can also be employed in the solution using finite element programs.

Finally, the ultimate in a CAD system such as this, is to provide a total system, which has a sound graphics base, a complete mesh data generation module, a solution module and finally a post processor. It is envisaged that such a system based on a minicomputer is not something of the future but of the present.

REFERENCES

1. Besant, C B.
"Computer Aided Design and Manufacture"
ELLIS HORWOOD LTD
pp 12-15 (1980).
2. Newman, W M and Sproull, R F.
"Principles of Interactive Computer Graphics"
McGRAW-HILL INC 1979.
3. Sutherland I E.
"Sketchpad: A Man-Machine Graphical Communication System"
PhD Thesis
MIT 1962.
4. Zienkiewicz O C.
"The Finite Element Method in Engineering Science"
McGRAW-HILL
LONDON 1971.
5. Turner M J et al.
"Stiffness and deflection analysis of complex structures"
J. Aero. Sci. 1956, 23, 805-823.
6. Crisp R J
"FEMG-Mesh generation program to fit triangular elements
in a uniform or non-uniform manner within a quadrilateral"
C.E.G.B. Report RD/C/N309 1969.
7. Latham D B
"A User's Guide to SAFECT"
C.E.G.B. Report RD/B/N1759 1970
8. Neale B K
"BERGEN - An Automatic Finite Element Mesh Generation
Program for Arbitrary Structures"
C.E.G.B. Report RD/B/N2432 1972.
9. Ergatoudis J, Irons B M and Zienkiewicz
"Curved isoparametric quadrilateral elements for Finite
Element Analysis"
Int. J. Solid Structures, 4, 31-42 (1968)
10. Ghassemi F.
"A General Data Preparation System for Finite Element
Methods using CAD techniques"
PhD Thesis, University of London, 1978.
11. Zienkiewicz O C and Phillips D V.
"An Automatic mesh generation scheme for plane and curved
surfaces by isoparametric coordinates"
Int. Journal for Numerical Methods in Engineering,
Vol 3 519-528 1971.

12. Hamlyn A.
"The Application of CAD Techniques to Building Engineering Design"
PhD Thesis, University of London 1974.
13. Ferner T Roger.
"Computing for Engineers"
The MACMILLAN PRESS LTD
LONDON 1974.
14. PDP 11/45 Processor Handbook 1978/79
Digital Equipment Corporation
Maynard Massachusetts (1978)
15. CAMAC - a Modern Instrumentation System for Data Handling
Euratom, EUR4100E (1964).
16. QUADRASYNC - A user's guide
ABLE COMPUTER TECHNOLOGY, Inc (1977).
17. DOS/BATCH Handbook 1974/1975
Digital Equipment Corporation
Maynard Massachusetts (1974).
18. PAK H A.
"Microprocessors in Computer Aided design and in Computer Aided Manufacture in Mechanical Engineering"
PhD Thesis, Imperial College, LONDON 1981.
19. Yi C.
"The use of Computer Aided Design Techniques in Dynamic Graphical Simulation"
PhD Thesis, Imperial College, LONDON 1974.
20. Besant C B and Jebb A.
"CADMAC-11, a Fully Interactive Computer Aided Design System"
Journal of CAD Vol 4 No. 5 (1972).
21. Ferner T Roger.
"Finite Element Methods for Engineers"
The MACMILLAN PRESS LTD, LONDON 1975.
22. Collins R J.
"Bandwidth reduction by automatic renumbering"
International Journal for Numerical Methods in Engineering
Vol 6 345-356 (1973)
23. Thompson D.
"The application of Computer Aided Draughting to Mechanical Engineering Design"
PhD Thesis, Imperial College, LONDON, June 1981.

24. McClintock P.
"The application of Computer Aided Design in Structural engineering draughting"
DIC Dissertation, Imperial College 1974.
25. Yi C, Besant C B and Jebb A.
"Three dimensional Digitising Techniques for Computer Aided animation"
CAD 76 Proceeding 1976.
26. Zienkiewicz O C, Irons B M, Ergatoudis J, Ahmad S and Scott F C.
"Isoparametric and associated Element Families for Two and Three Dimensional Analysis"
Tapir Press
Trondheim 1969 Chapter 13.
27. Schrem E.
"ASKA Part 1 - Linear Static Analysis, User's Reference Manual"
ISD-Report No. 73 (ASKA UM 202), University of Stuttgart (1971).
28. Henshell R D.
"PAFEL 75 EASIDATA"
PAFEC HOUSE, PAFEC Ltd, Nottingham 1977.
29. D Hitchings and F L Mathews
"FINEL - Finite Elements in Theory and Practice"
Aeronautical Engineering Department
Imperial College, LONDON 1978.

APPENDIX A FLOW CHART

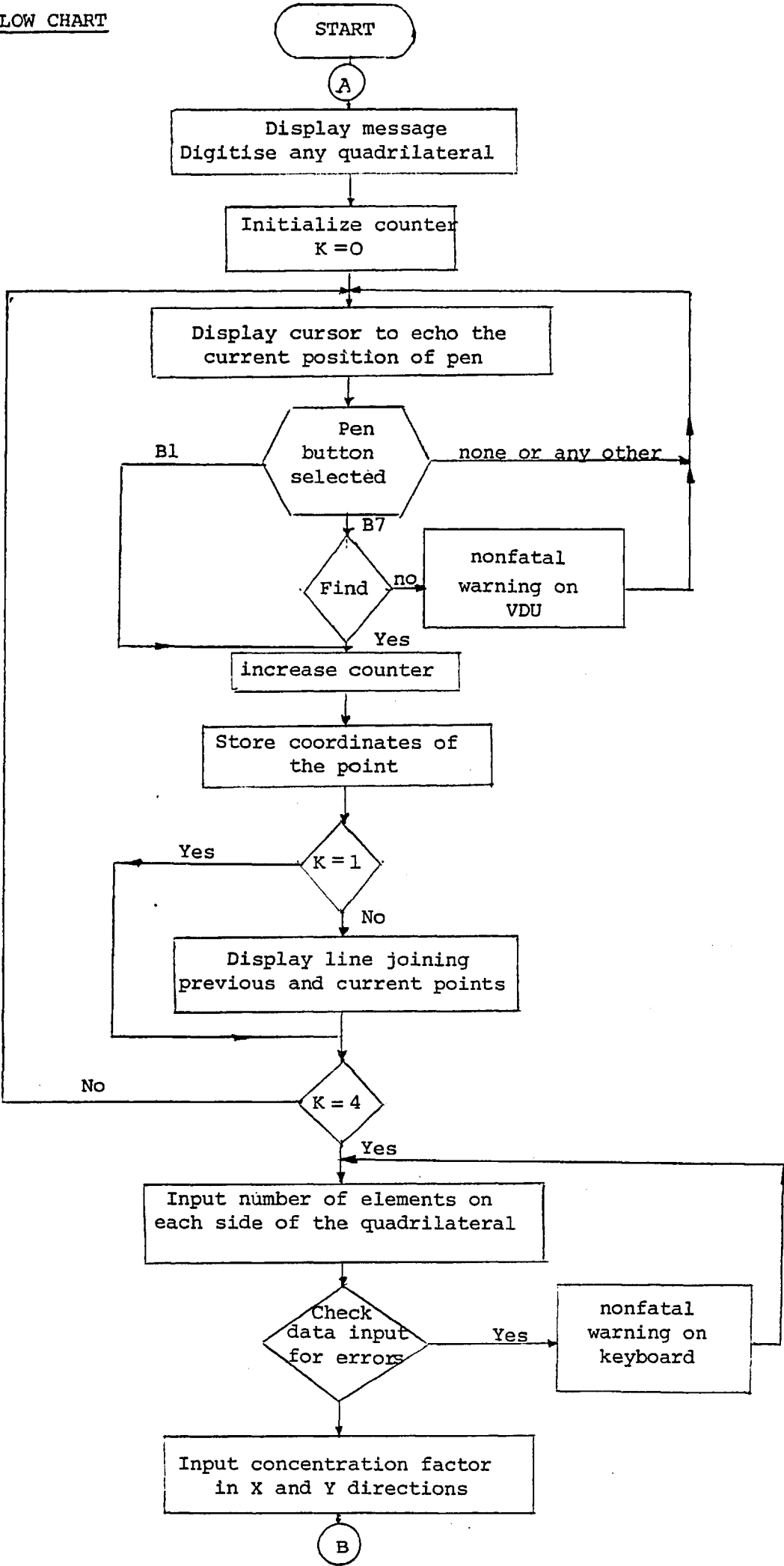


Fig A1.1
FLOW CHART FOR
MESH 2D

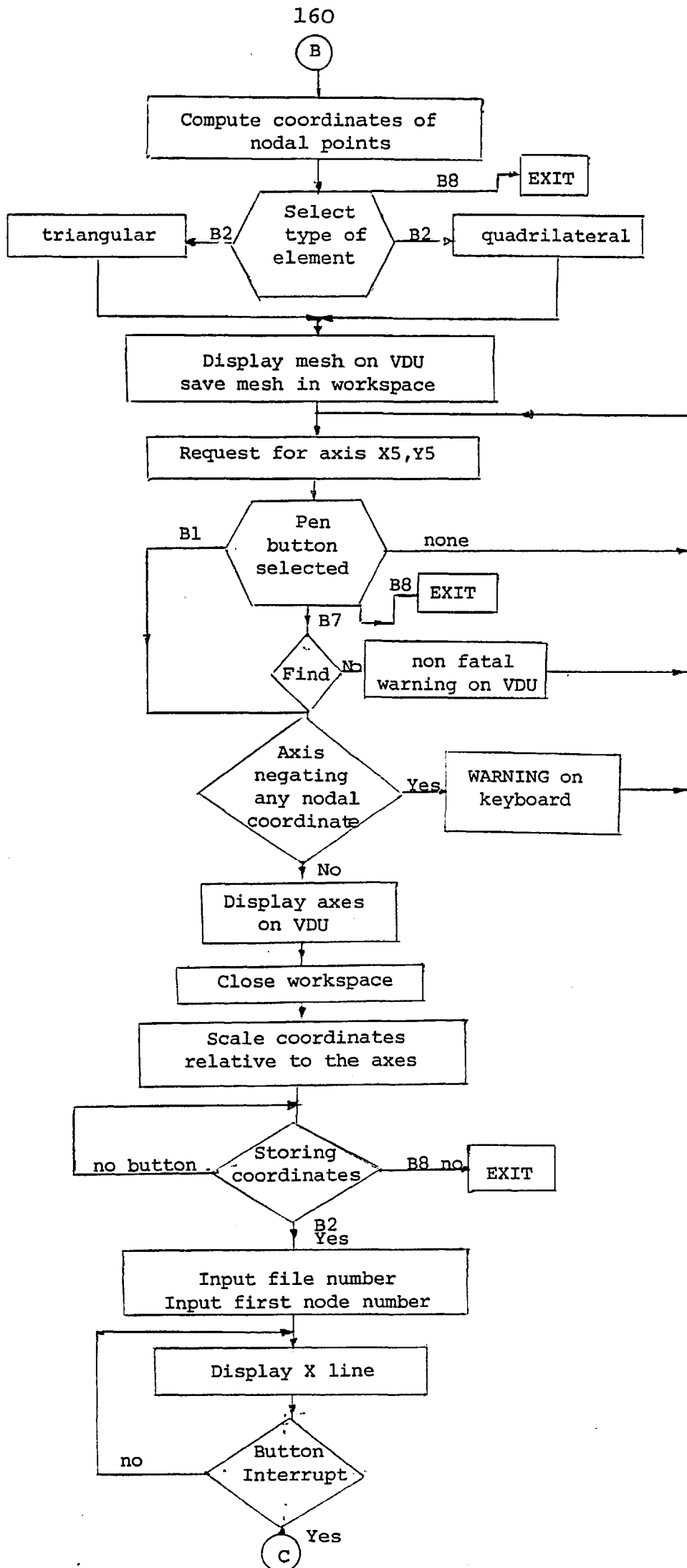


Fig A1.2

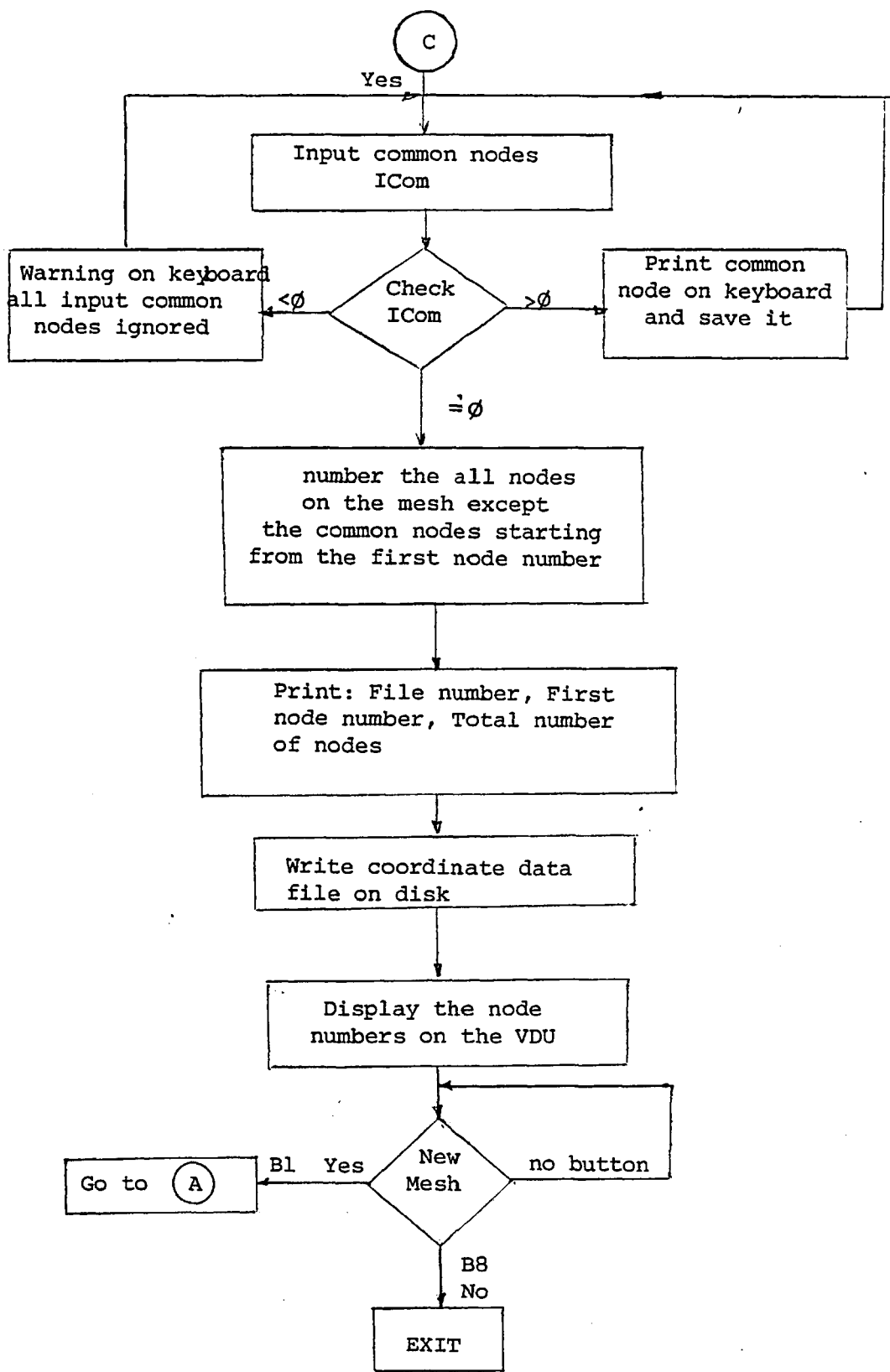


Fig A1.3

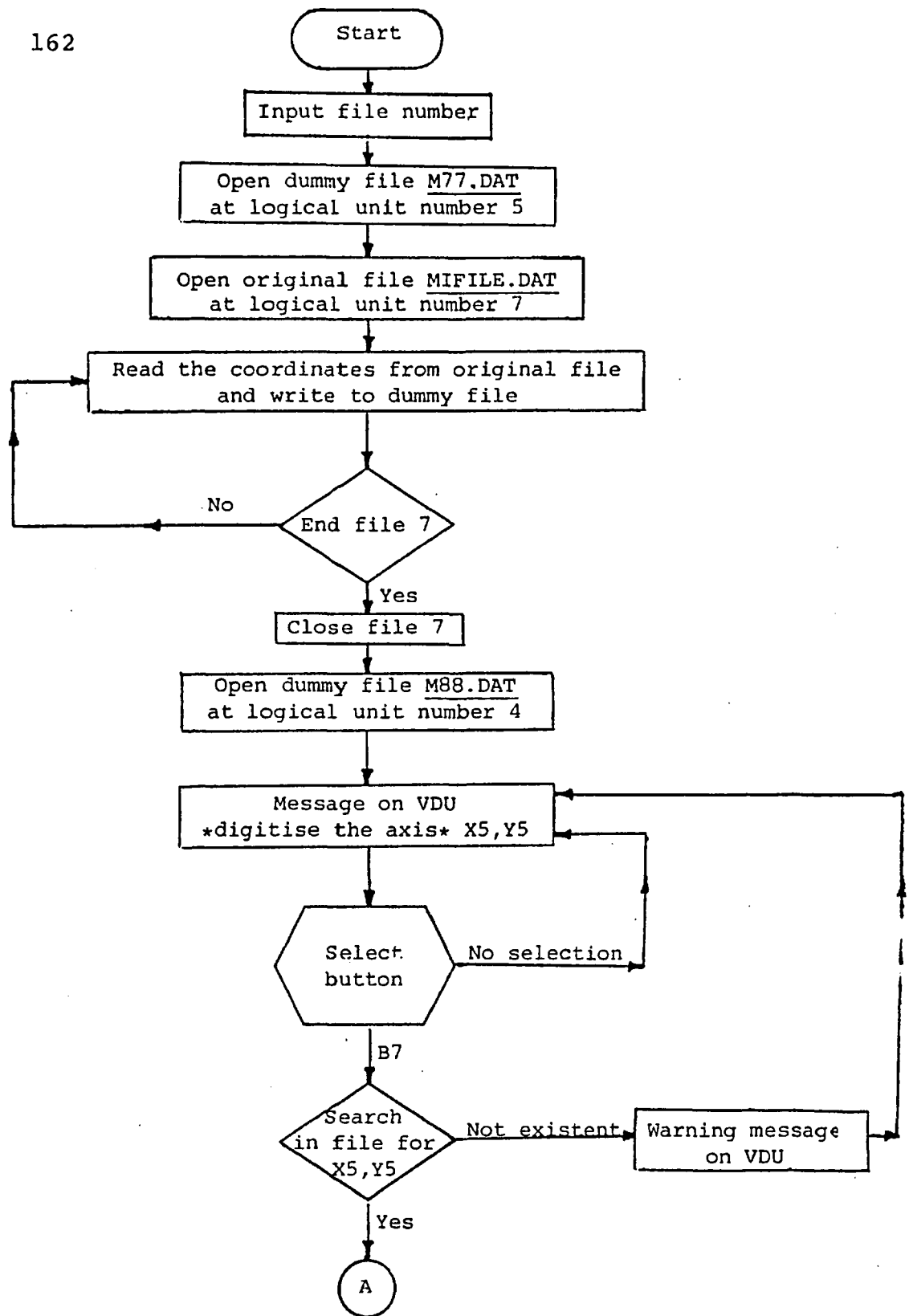


Fig A2.1 Outline Flowchart of ADDELM

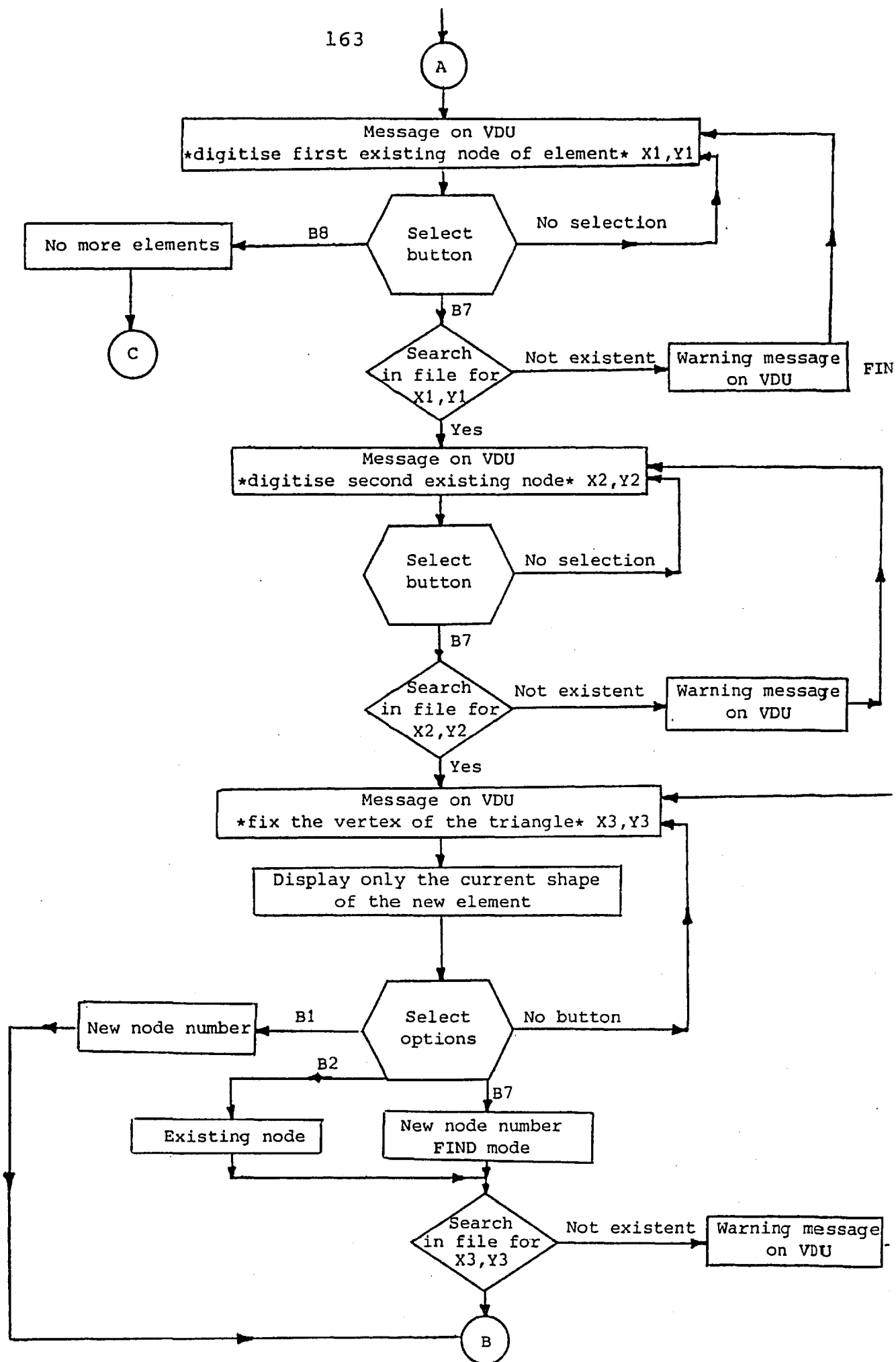


Fig A2.2

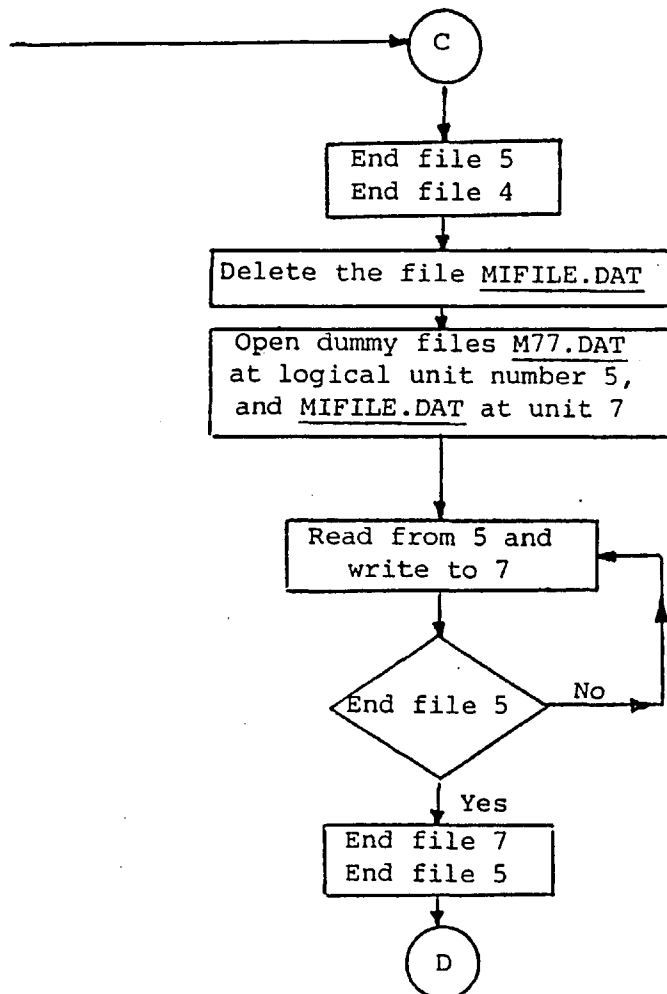
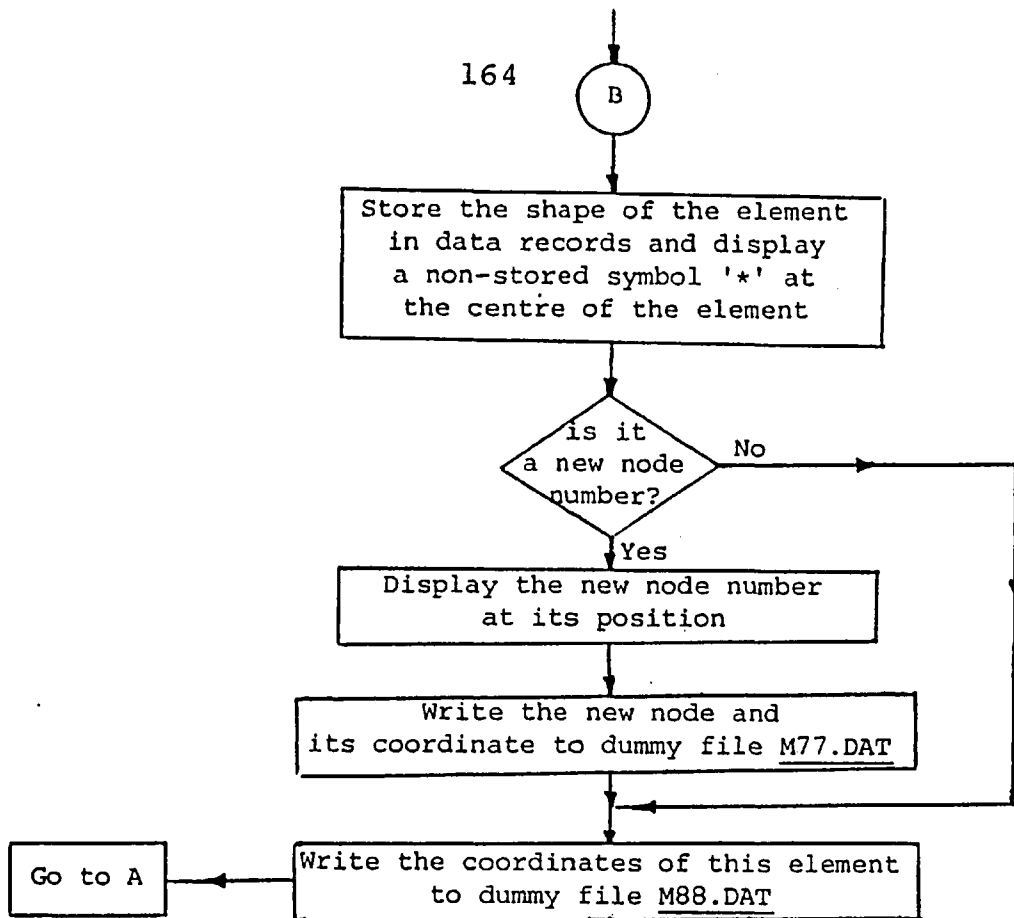


Fig A2.3

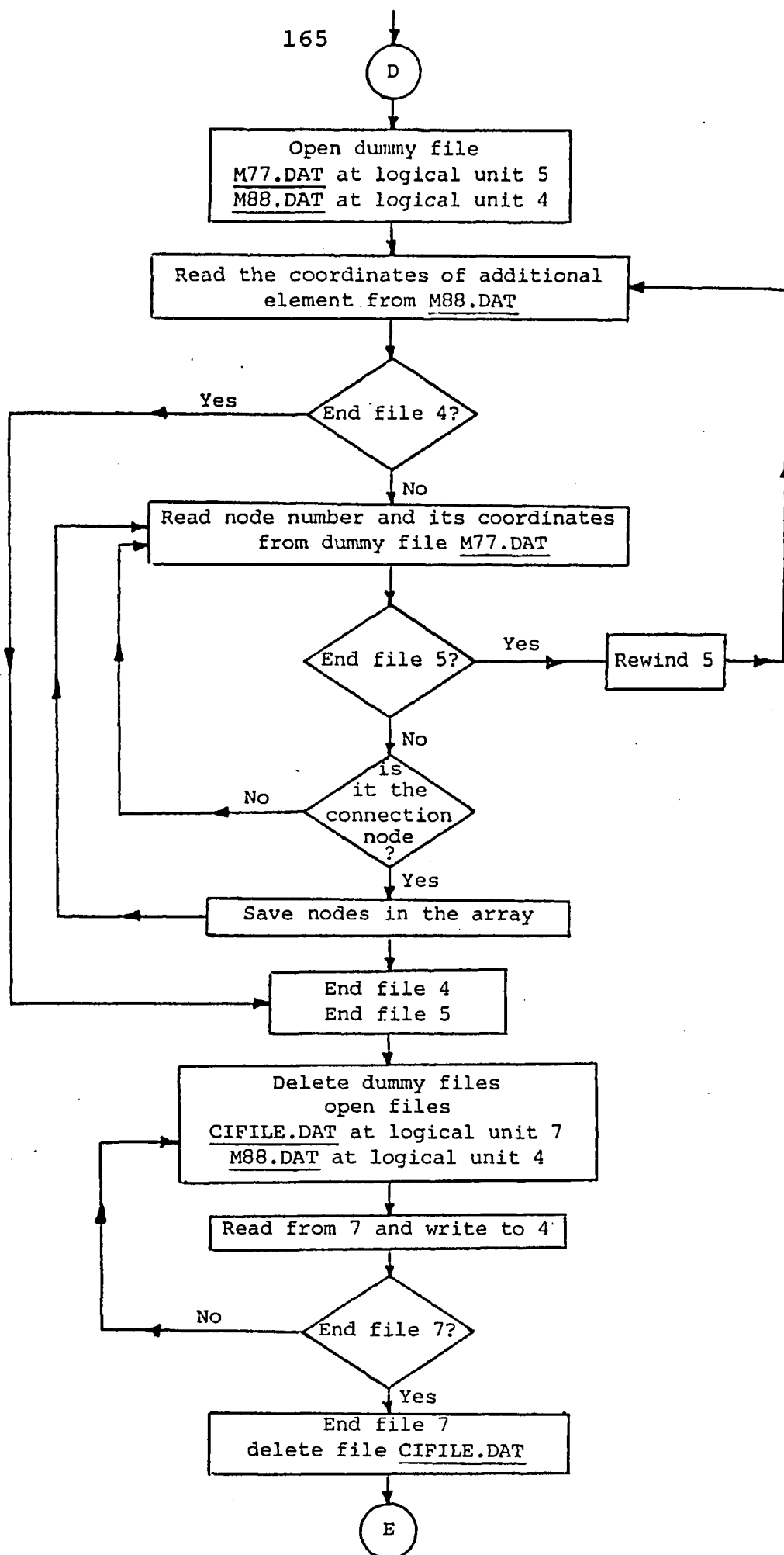


Fig A2.4

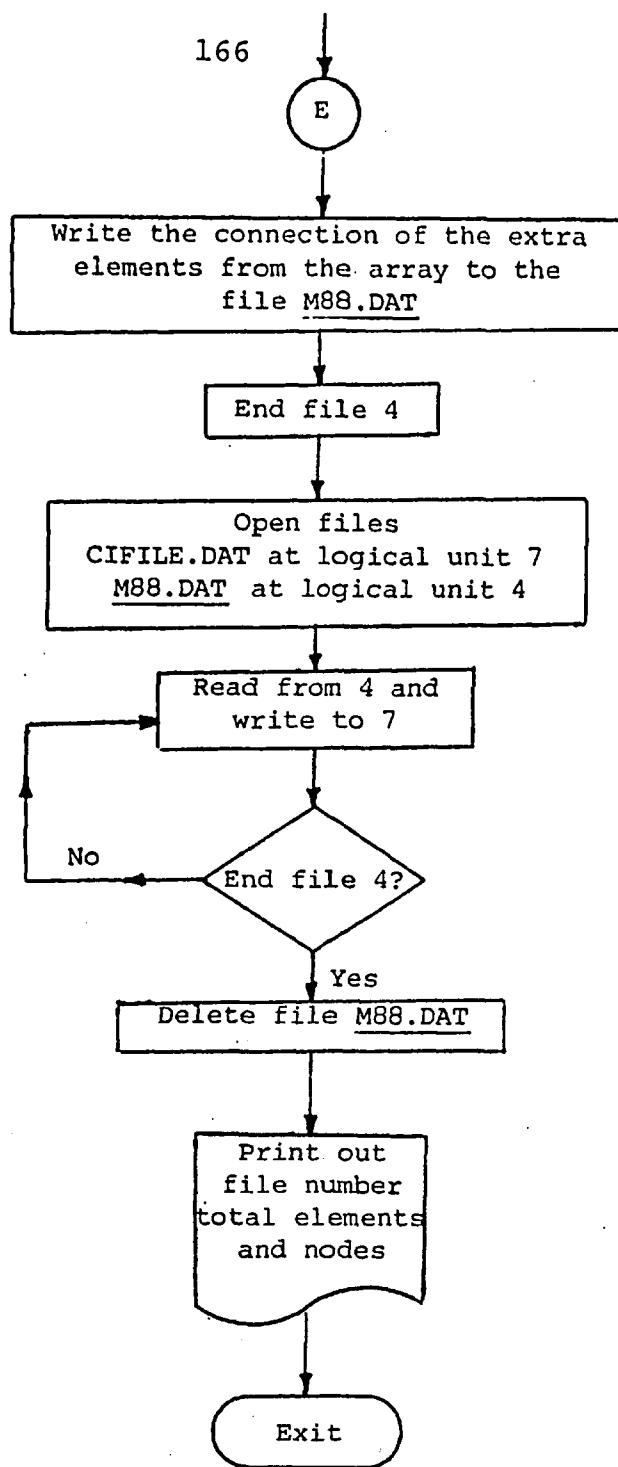


Fig A2.5

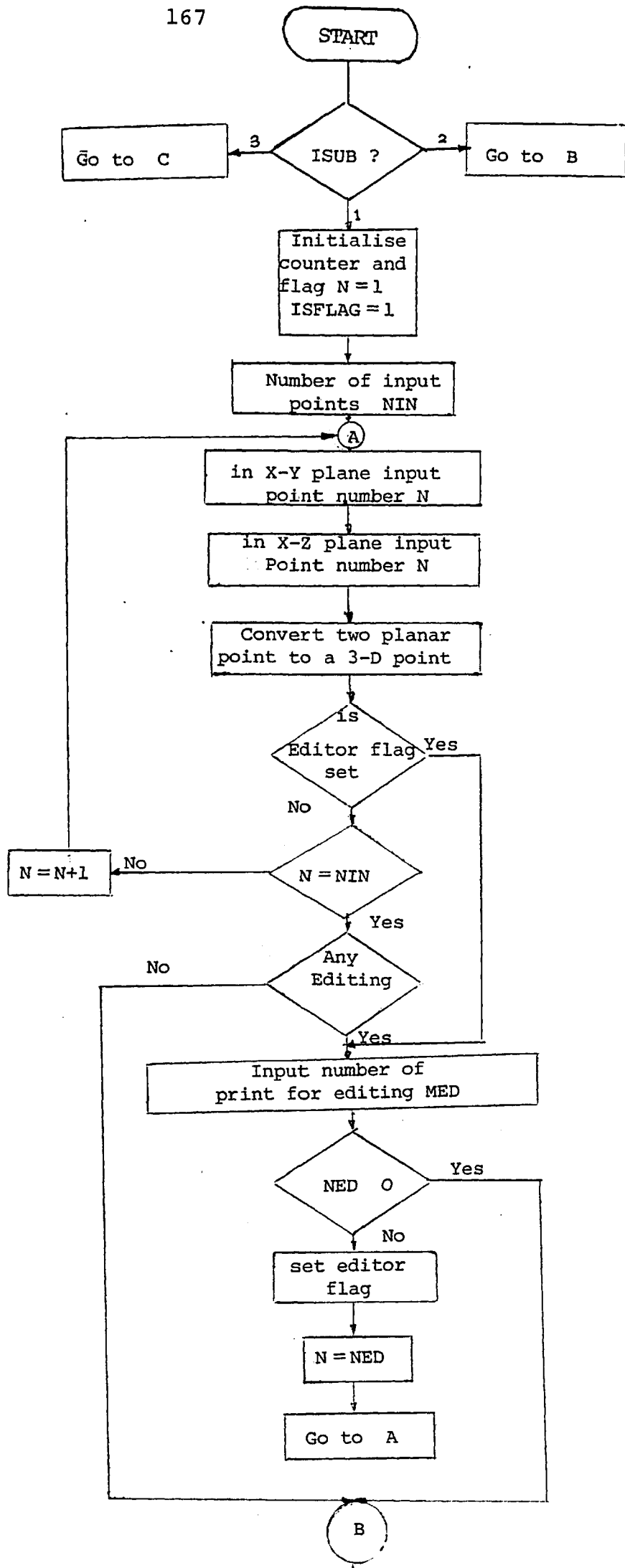


Fig A321

FLOW CHART FOR
3D MESH

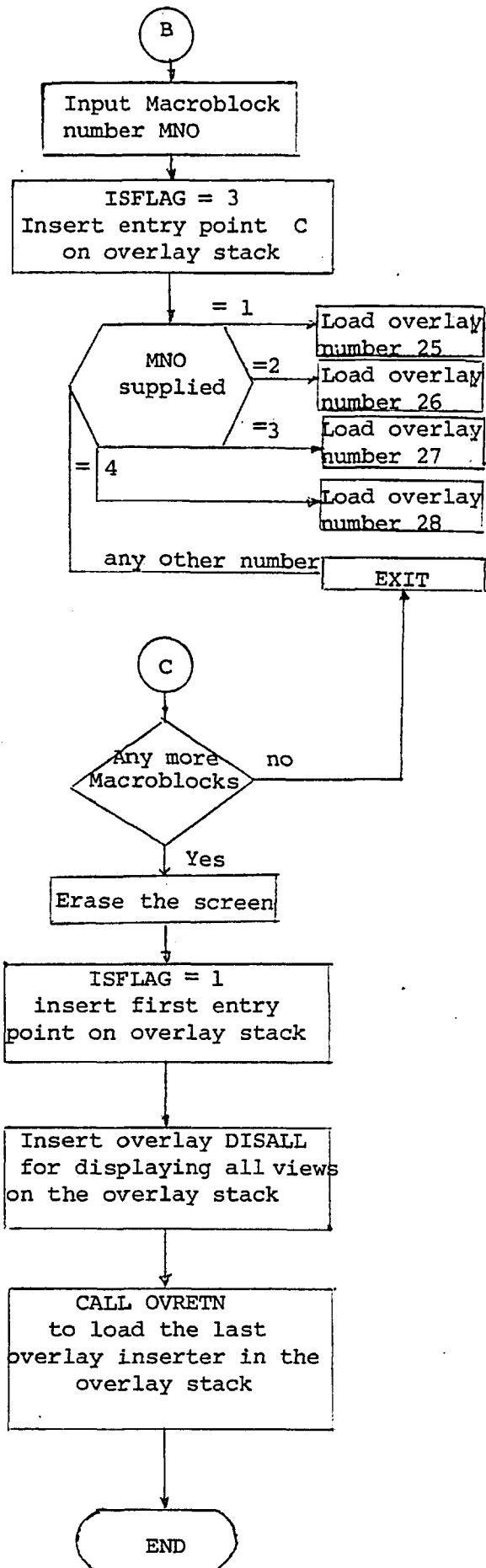


Fig A3.2

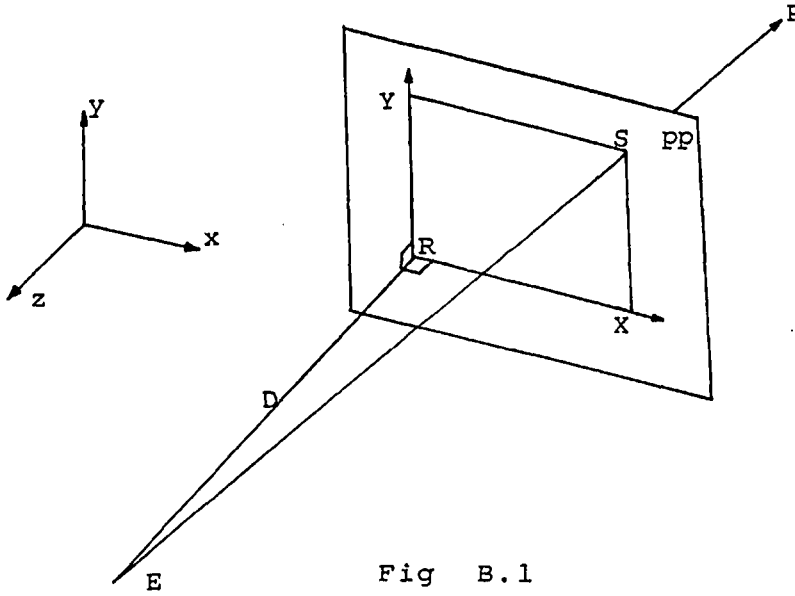
APPENDIX BPERSPECTIVE

Fig B.1

Let O be the origin of the spatial cartesian coordinate system

Let PP be defines as a projection plane with coordinates X, Y

Let the observers eye be situated at a point $\bar{E} (x_e, y_e, z_e)$

and the line of sight making angles α, β , and γ with the spatial x, y and z axes respectively.

Let the plane of projection PP be at a distance D from $\bar{E}$,

normal to the line of sight, $\bar{R} [x_r, y_r, z_r]$ being the normal from $\bar{E}$ to the plane Fig B1

the scalar D is given by the following equation

$$D = \overline{ER} = \left[(x_e - x_r)^2 + (y_e - y_r)^2 + (z_e - z_r)^2 \right]$$

and the direction cosines

$$\cos \alpha = (x_e - x_r) / D$$

$$\cos \beta = (y_e - y_r) / D$$

$$\cos \gamma = (z_e - z_r) / D$$

$\bar{S} [x_s, y_s, z_s]$ is the perspective image of the point $P(x_p, y_p, z_p)$ in the plane PP with respect to $\bar{E}$.

To generate the perspective view of an object defined by points like $\bar{P}$, the spatial coordinates (x_p, y_p, z_p) must be converted to (X, Y) coordinates of plane PP .

Referring to Fig B1 from vector algebra

$$\bar{R} = \bar{E} + D (\cos \alpha, \cos \beta, \cos \gamma) \quad (1)$$

From geometry the equation of the straight line ESP is given by

$$\frac{x_s - x_e}{x_p - x_e} = \frac{y_s - y_e}{y_p - y_e} = \frac{z_s - z_e}{z_p - z_e} = K \quad (2)$$

where K is a constant

$$\text{From 1 } \bar{ER} = \bar{R} - \bar{E} = D[\cos \alpha, \cos \beta, \cos \gamma] \quad (3)$$

Since $\bar{S}$ lies a plane PP

$$\bar{ER} \cdot \bar{RS} = 0 \quad (4)$$

but $\bar{RS} = \bar{ES} - \bar{ER}$

therefore $\bar{ER} \cdot (\bar{ES} - \bar{ER}) = 0$

$$\bar{ER} \cdot \bar{ES} = \bar{ER} \cdot \bar{ER} = D^2 \quad (5)$$

$$\bar{ES} = [x_s - x_e, y_s - y_e, z_s - z_e]$$

From (3) and (5)

$$D (\cos \alpha, \cos \beta, \cos \gamma) [x_s - x_e, y_s - y_e, z_s - z_e] = D^2$$

$$\text{therefore } (x_s - x_e) \cos \alpha + (y_s - y_e) \cos \beta + (z_s - z_e) \cos \gamma = D$$

$$\rightarrow K = D / [(x_p - x_e) \cos \alpha + (y_p - y_e) \cos \beta + (z_p - z_e) \cos \gamma] \quad (6)$$

from which K can be calculated since all other terms are known $\bar{S}$ can be determined From (2)

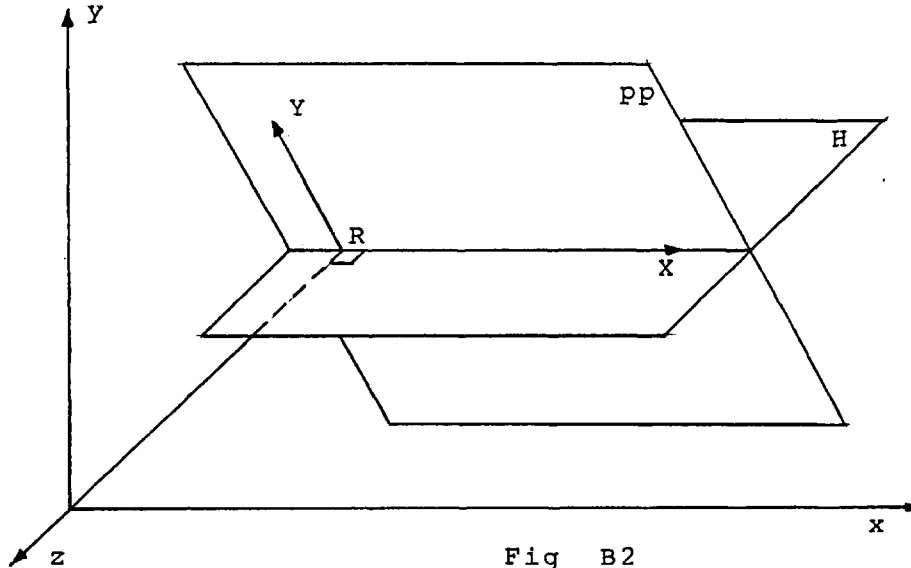
$$\bar{S} = \bar{E} + K(\bar{P} - \bar{E}) \quad (7)$$

Next, express $\bar{S}$ in terms of the two-dimensional system of coordinates in PP[X,Y] . The equation of PP is given by (4)

$$(x_s - x_r) \cos \alpha + (y_s - y_r) \cos \beta + (z_s - z_r) \gamma = 0 \quad (8)$$

Consider the horizontal plane H in Fig B2 given by

$$y - y_r = 0$$



The intersection of planes PP and H is taken along the X direction

Equation of PP

$$x_s \cos \alpha + y_s \cos \beta + z_s \cos \gamma = x_r \cos \alpha + y_r \cos \beta + z_r \cos \gamma \quad (9)$$

Equation for H

$$y - y_r = 0$$

Let x, y, z be a point on the line of intersection

$$y = y_r$$

from (9)

$$x \cos \alpha + z \cos \gamma = x_r \cos \alpha + z_r \cos \gamma \quad (10)$$

$$x = x_r + (z_r - z) \frac{\cos \gamma}{\cos \alpha}$$

$$\frac{x - x_r}{\cos \alpha} = \frac{y - y_r}{\cos \beta} = \frac{z - z_r}{\cos \alpha}$$

A vector in the X direction is given by $[\cos \gamma, 0, -\cos \alpha]$

the unit vector in X direction U_x is given by

$$U_x = \pm \left[\frac{\cos \gamma}{\cos^2 \alpha + \cos^2 \gamma}, 0, \frac{-\cos \alpha}{\cos^2 \alpha + \cos^2 \gamma} \right] \quad (11)$$

$$\text{But } \cos^2 \alpha + \cos^2 \gamma = 1 - \cos^2 \beta - \sin^2 \beta$$

$$\begin{aligned} \text{therefore } \bar{U}_x &= \pm \left[\frac{\cos \gamma}{\sin \beta}, 0, \frac{-\cos \alpha}{\sin \beta} \right] \\ &= \lambda \left[\frac{\cos \gamma}{\sin \beta}, 0, \frac{\cos \alpha}{\sin \beta} \right] \end{aligned} \quad (12)$$

λ being ± 1 whose limits will be defined later,

Let the unit vector in the y direction be

$$\bar{U}_y = [a, b, c] \quad (13)$$

$\bar{U}_y$ must satisfy the following equation

$$\bar{U}_x \bar{U}_y = 0$$

$$(\bar{U}_y) = 1$$

$$\bar{U}_y \text{ OR } = 0$$

$$a \cos \gamma - c \cos \alpha = 0 \quad (14)$$

$$a^2 + b^2 + c^2 = 1 \quad (15)$$

$$a \cos \alpha + b \cos \beta + c \cos \gamma = 0 \quad (16)$$

From (14) $a = c \frac{\cos \alpha}{\cos \gamma}$

substitute in (16)

$$c \frac{\cos^2 \alpha}{\cos \gamma} + c \cos \gamma = -b \cos \beta$$

$$b = -c \frac{\cos^2 \alpha + \cos^2 \gamma}{\cos \beta \cos \gamma} = -c \frac{\sin^2 \beta}{\cos \beta \cos \gamma}$$

Substitute in (15)

$$c^2 \frac{\cos^2 \alpha}{\cos^2 \gamma} + c^2 + c \frac{\sin^4 \beta}{\cos^2 \beta \cos^2 \gamma} = 1$$

$$c^2 = \frac{\cos^2 \beta \cos^2 \gamma}{\sin^2 \beta}$$

$$c = \rho \frac{\cos \beta \cos \gamma}{\sin \beta}$$

when $\rho = \pm 1$

substituting back

$$a = \rho \frac{\cos \alpha \cos \beta}{\sin \beta}$$

$$b = -\rho \sin \beta$$

$$\bar{U}_y = \rho \left[\frac{\cos\alpha \cos\beta}{\sin\beta}, -\sin\beta, \frac{\cos\beta \cos\gamma}{\sin\beta} \right] \quad (17)$$

The values of λ and ρ will both be determined

For λ upwards

$$\bar{U}_y \cdot \mathbf{j} > 0$$

where $\mathbf{j}$ is the unit vector in the y direction

$$-\rho \sin\beta > 0$$

similarly for y downward

$$\rho \sin\beta < 0$$

$$\sin 0 < \beta < \pi$$

$$0 < \sin\beta < 1$$

$$\text{then } \rho = -1$$

$\bar{U}_x \times \bar{U}_y$ is normal to plane PP hence parallel to the line of sight and in the opposite direction

$$\begin{aligned} \bar{U}_x \times \bar{U}_y &= -\bar{O}R/D = -[\cos\alpha, \cos\beta, \cos\gamma] \\ &= \lambda\rho \left[\frac{\cos\gamma}{\sin\beta} \bar{i} + 0\bar{j} - \frac{\cos\alpha}{\sin\beta} \bar{k} \right] \times \\ &\quad \left[\frac{\cos\alpha \cos\beta}{\sin\beta} \bar{i} - \sin\beta \bar{j} + \frac{\cos\beta \cos\gamma}{\sin\beta} \bar{k} \right] \end{aligned} \quad (18)$$

Consider the $\bar{i}$ term

$$\mathbf{j} \times \bar{k} \cos\alpha \lambda\rho = -\cos\alpha \bar{i}$$

$$\lambda\rho = -1$$

$$\lambda = 1$$

Substituting for the values of λ and ρ in (12) and (17)

$$\bar{U}_x = \left[\frac{\cos\gamma}{\sin\beta}, 0 - \frac{\cos\alpha}{\sin\beta} \right] \quad (19)$$

$$\bar{U}_y = \left[-\frac{\cos\alpha\cos\beta}{\sin\beta}, \sin\beta, \frac{\cos\beta\cos\gamma}{\sin\beta} \right]$$

Components of $\bar{S}$ in terms of the new axes will be

$$X = \bar{R}\bar{S} \bar{U}_x = \left((x_s - x_r) \cos\gamma - (z_s - z_r) \cos\alpha \right) / \sin\beta \quad (20)$$

$$Y = \bar{R}\bar{S} \bar{U}_y = - (x_s - x_r) \frac{\cos\alpha\cos\beta}{\sin\beta} - (z_s - z_r) \frac{\cos\gamma\cos\beta}{\sin\beta} \quad (21)$$

From (8)

$$(x_s - x_r) \cos\alpha + (z_s - z_r) \cos\gamma = - (y_s - y_r) \cos\beta$$

$$\text{which gives } Y = (y_s - y_r) / \sin\beta \quad (22)$$

Remembering the following

Eye position $\bar{E} [x_e, y_e, z_e]$

direction of sight $[\alpha, \beta, \gamma]$

distance from PP = D

coordinate of P $[x_p, y_p, z_p]$

Compute:

$$\begin{aligned} \text{I} \quad x_r &= x_e + D \cos\alpha \\ y_r &= y_e + D \cos\beta \\ z_r &= z_e + D \cos\gamma \end{aligned}$$

$$\text{II} \quad K = D / [(x_p - x_e) \cos\alpha + (y_p - y_e) \cos\beta + (z_p - z_e) \cos\gamma]$$

$$\begin{aligned} \text{III} \quad x_s &= x_e + K(x_p - x_e) \\ y_s &= y_e + K(y_p - y_e) \\ z_s &= z_e + K(z_p - z_e) \end{aligned}$$

$$X = (x_s - x_r) \cos \gamma - (z_s - z_r) \cos \alpha / \sin \beta$$

$$Y = (y_s - y_r) / \sin \beta$$

The above equations are acceptable in any perspective case except when $\sin \beta = 0$ i.e. $\beta = 0, \pi$ when the last two equations degenerate. The line of sight is vertical and the above transformation cannot be used. PP coincides with the plane $y - y_r = 0$. For such cases the X axis can be defined by the intersection of PP and $z - z_r = 0$ for which the singular transformation can be used (Fig B3):

$$X = x_s - x_r$$

$$Y = (z_r - z_s) \cos \beta$$

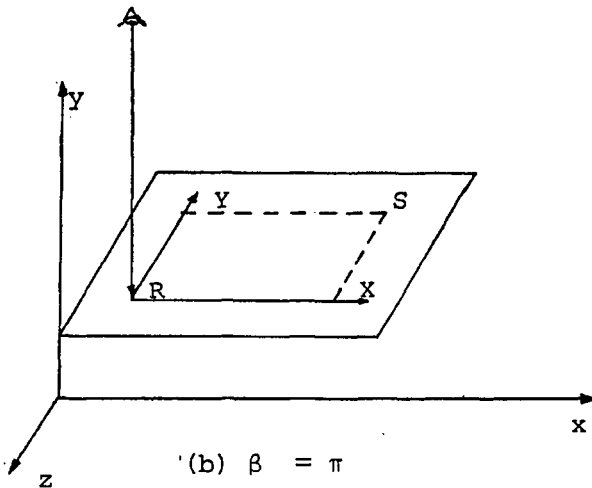
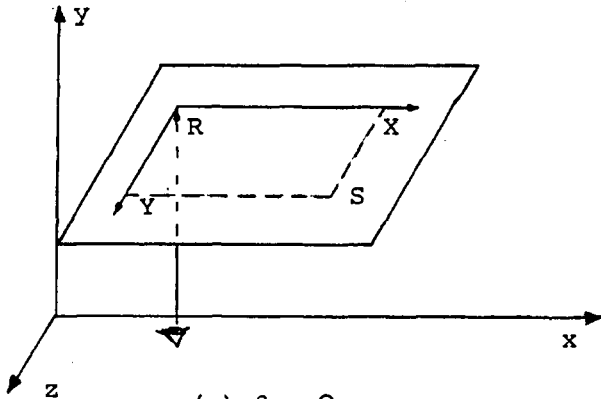


Fig B3 Similar Transformations

APPENDIX C

```

C
C   INPUT THE NUMBER OF ELEMENTS ON EACH SIDE
832   CALL BUTNUM('ELEMENTS IN X',NXL,UM)
      CALL BUTNUM('ELEMENTS IN Y',NYL,UM2)
      NNP=(NXL+1)*(NYL+1)
C **CHECK THE INPUT DATA SUPPLIED BY THE USER**
      IF(NX.LE.0.OR.NYL.LE.0)GO TO 833
      IF (NNP.GT.256)GO TO 833
C **DEFINE THE FOUR MAJOR LENGHTS**
      XB=CX(4)-CX(1)
      XT=CX(3)-CX(2)
      YL=CY(2)-CY(1)
      YR=CY(3)-CY(4)
C
      NXP=NXL+1
      NYP=NYL+1
      DO 2 IY=1,NYP
        DO 2 IX=1,NXP
C   COMPUTE THE NODE COUNTER
          I=(IY-1)*NXP+IX
          XIR=FLOAT(IX-1)/FLOAT(NXL)
          YIR=FLOAT(IY-1)/FLOAT(NYL)
C   GET COORDINATES
          X(I)=CX(1)+XIR*XB+(CX(2)+XIR*XT-XIR*XB-CX(1))*YIR
          Y(I)=CY(1)+YIR*YL+(CY(4)+YIR*YR-YIR*YL-CY(1))*XIR
2     CONTINUE

```

Listing 4.1

Program section to generate coordinates using
coordinate geometric algorithm

```

C
C INPUT THE NUMBER OF ELEMENTS ON EACH SIDE
832 CALL BUTNUM('ELEMENTS IN X',NXL,UM)
CALL BUTNUM('ELEMENTS IN Y',NYL,UM2)
NNP=(NXL+1)*(NYL+1)

C **CHECK THE DATA INPUT BY THE USER**
IF(NXL.LE.0.OR.NYL.LE.0)GO TO 833
IF(NNP.GT.256)GO TO 833

C**DEFINE THE COORDINATE OF EACH NODE**
XB=CX(4)-CX(1)
XT=CX(3)-CX(2)
YL=CY(2)-CY(1)
YR=CY(3)-CY(4)
NXP=NXL+1
NYP=NYL+1
830 CALL BUTNUM('CONTRATION FACTOR IN X',COX,COX)
CALL BUTNUM('CONTRATION FACTOR IN Y',COY,COY)

C THE CONTRATION FACTOR MUST NOT BE ZERO OR NEGATIVE
IF(COX.LE.0.0)GO TO 830
IF(COY.LE.0.0)GO TO 830
DO 2 IY=1,NYP
DO 2 IX=1,NXP
C DEFINE THE NODE NUMBER COUNTER
I=(IY-1)*NXP+IX
XIR=FLOAT(IX-1)/FLOAT(NXL)
YIR=FLOAT(IY-1)/FLOAT(NYL)
IF(COX.NE.1.)XIR=(COX**((IX-1)-1.)/((COX**(NXL)-1.))
IF(COY.NE.1.)YIR=(COY**((IY-1)-1.)/((COY**(NYL)-1.0)
X(1)=CX(1)+XIR*XB+(CX(2)+XIR*XT-XIR*XB-CX(1))*YIR
Y(1)=CY(1)+YIR*YL+(CY(4)+YIR*YR-YIR*YL-CY(1))*XIR
2 CONTINUE

```

Listing 4.2

Program section for generating uniform and concentrated meshes in MESH2D

MODULE GRID

GENERATED MESH CO-ORDINATES

NODAL POINT	X CO-ORD	Y CO-ORD	Z CO-ORD	NODAL POINT	X CO-ORD	Y CO-ORD	Z CO-ORD	NODAL POINT	X CO-ORD	Y CO-ORD	Z CO-ORD
1	1396.234	32223.514	1418.768	2	1424.682	32224.752	1418.121	3	1453.892	32224.784	1417.772
4	1483.864	32223.514	1417.740	5	1514.598	32221.228	1419.024	6	1468.331	32245.533	1419.010
7	1436.226	32246.120	1417.534	8	1509.581	32444.437	1417.644	9	1417.576	32267.964	1419.042
10	1437.303	32266.789	1418.010	11	1457.064	32266.186	1417.454	12	1478.573	32266.154	1417.375
13	1496.310	32265.694	1417.772	14	1423.952	32267.808	1418.833	15	1456.226	32284.982	1417.534
16	1456.785	32267.998	1418.407	17	1427.476	32265.064	1418.534	18	1441.097	33003.111	1417.836
19	1453.892	32202.308	1417.772	20	1465.862	32264.354	1418.344	21	1477.066	33003.350	1419.550
22	1419.772	32223.514	1554.015	23	1479.045	32224.752	1554.015	24	1542.987	32221.228	1555.874
25	1440.896	32267.984	1554.283	26	1482.107	32266.186	1553.485	27	1523.911	32200.604	1554.678
28	1452.284	32265.064	1553.602	29	1477.963	32202.308	1553.803	30	1505.059	32266.350	1556.315
31	1444.917	32223.514	1553.133	32	1474.173	32224.752	1553.720	33	1504.951	32224.784	1553.744
34	1533.015	32223.514	1553.133	35	1558.072	32221.228	1552.403	36	1456.160	32245.533	1553.744
37	1506.469	32246.120	1553.312	38	1559.711	32244.437	1553.864	39	1465.467	32267.964	1553.744
40	1485.229	32266.789	1553.061	41	1507.380	32266.186	1553.095	42	1526.859	32266.154	1553.744
43	1550.656	32265.694	1553.312	44	1472.535	32267.808	1553.104	45	1507.223	32284.982	1553.142
46	1541.536	32267.998	1553.312	47	1477.653	32265.064	1553.804	48	1442.158	33003.111	1553.655
49	1506.013	32265.064	1553.455	50	1549.403	32202.308	1553.605	51	1532.322	33003.350	1553.306
52	1471.672	32223.514	1553.490	53	1536.409	32224.752	1553.103	54	1544.354	32221.228	1553.614
55	1471.103	32267.984	1553.474	56	1532.222	32266.186	1553.283	57	1576.274	32266.154	1553.222
58	1504.184	32265.694	1553.966	59	1532.061	32202.308	1553.722	60	1558.794	33003.350	1553.522
61	1500.036	32223.514	1553.638	62	1527.587	32224.752	1553.971	63	1555.621	32224.784	1553.961
64	1567.136	32223.514	1553.606	65	1619.133	32221.228	1553.908	66	1509.592	32245.533	1553.915
67	1557.797	32246.120	1553.320	68	1610.342	32244.437	1553.160	69	1517.985	32267.964	1553.425
70	1537.855	32266.789	1553.956	71	1556.434	32266.186	1553.750	72	1579.691	32266.154	1553.704
73	1601.635	32265.694	1553.920	74	1525.213	32287.308	1553.167	75	1558.531	32284.982	1553.150
76	1593.014	32267.998	1553.188	77	1531.278	32265.064	1553.143	78	1544.736	33003.112	1553.149
79	1598.089	32265.064	1553.622	80	1571.335	32202.308	1553.540	81	1584.477	33003.350	1553.644
82	1530.060	32223.514	2105.576	83	1583.185	32224.752	2104.344	84	1543.009	32221.229	2105.384
85	1546.052	32267.984	2105.278	86	1584.215	32266.186	2103.394	87	1625.350	32266.154	2105.242
88	1559.133	32265.694	2105.334	89	1584.102	32202.308	2104.134	90	1609.369	32266.350	2105.633
91	1551.588	32223.514	2244.304	92	1584.924	32224.752	2243.375	93	1610.102	32224.784	2243.277
94	1518.121	32267.998	2244.394	95	1625.982	32221.228	2244.542	96	1586.605	32245.533	2244.447
97	1518.150	32246.120	2244.557	98	1625.982	32244.437	2243.525	99	1575.364	32267.964	2244.034
100	1518.150	32223.514	2244.375	101	1610.165	32266.186	2242.319	102	1629.168	32266.154	2242.367
103	1518.150	32267.998	2244.018	104	1581.686	32267.808	2244.056	105	1610.150	32284.982	2242.356
106	1644.121	32265.694	2244.018	107	1587.750	32265.064	2244.542	108	1598.749	33003.111	2243.717
109	1610.102	32265.064	2244.272	110	1621.659	32202.308	2243.209	111	1633.470	33003.350	2243.526
112	1633.470	32223.514	2244.272	113	1322.511	32212.751	1417.836	114	1352.546	32217.672	1419.298
115	1377.121	32221.228	1419.614	116	1322.130	32242.151	1419.360	117	1374.574	32245.602	1420.042
118	1352.546	32267.998	1421.328	119	1374.184	32272.146	1420.804	120	1392.360	32271.012	1419.042
121	1366.254	32265.694	1421.328	122	1376.740	32202.308	1421.138	123	1408.505	32245.533	1419.163
124	1396.234	32266.789	1421.328	125	1407.753	32266.186	1421.847	126	1416.868	32266.154	1419.518
127	1423.952	32265.694	1421.328	128	1312.847	32202.308	1557.275	129	1375.519	32217.672	1557.140
130	1423.952	32267.998	1421.328	131	1415.062	32271.011	1558.991	132	1419.772	32223.514	1554.955
133	1440.896	32265.694	1552.971	134	1336.431	32202.308	1557.734	135	1369.775	32217.672	1557.734
136	1420.896	32265.064	1552.971	137	1424.019	32221.228	1552.974	138	1370.255	32245.533	1552.974
139	1420.896	32223.514	1552.974	140	1400.411	32273.522	1552.974	141	1420.896	32272.345	1552.974
142	1420.896	32267.998	1552.974	143	1493.110	32269.350	1552.974	144	1420.896	32266.350	1552.974
145	1420.896	32265.694	1552.974	146	1493.110	32269.350	1552.974	147	1420.896	32266.350	1552.974
148	1420.896	32265.694	1552.974	149	1493.110	32269.350	1552.974	150	1420.896	32266.350	1552.974
151	1420.896	32265.694	1552.974	152	1493.110	32269.350	1552.974	153	1420.896	32266.350	1552.974
154	1420.896	32265.694	1552.974	155	1493.110	32269.350	1552.974	156	1420.896	32266.350	1552.974
157	1420.896	32265.694	1552.974	158	1493.110	32269.350	1552.974	159	1420.896	32266.350	1552.974
160	1420.896	32265.694	1552.974	161	1493.110	32269.350	1552.974	162	1420.896	32266.350	1552.974
163	1420.896	32265.694	1552.974	164	1493.110	32269.350	1552.974	165	1420.896	32266.350	1552.974
166	1420.896	32265.694	1552.974	167	1493.110	32269.350	1552.974	168	1420.896	32266.350	1552.974
169	1420.896	32265.694	1552.974	170	1493.110	32269.350	1552.974	171	1420.896	32266.350	1552.974
172	1420.896	32265.694	1552.974	173	1427.208	32217.672	2105.314	174	1476.969	32273.552	2105.153
176	1427.208	32233.514	2105.574	177	1455.608	32233.514	2105.574	178	1455.608	32233.514	2105.574
179	1427.208	32202.308	2105.574	180	1427.208	32212.751	2105.574	181	1427.208	32212.751	2105.574
182	1427.208	32246.120	2105.574	183	1427.208	32242.151	2105.574	184	1427.208	32242.151	2105.574
185	1427.208	32266.789	2105.574	186	1427.208	32266.186	2105.574	187	1427.208	32266.186	2105.574
188	1427.208	32266.789	2105.574	189	1427.208	32266.186	2105.574	190	1427.208	32266.186	2105.574
191	1427.208	32266.789	2105.574	192	1427.208	32266.186	2105.574	193	1427.208	32266.186	2105.574
194	1427.208	32266.789	2105.574	195	1427.208	32266.186	2105.574	196	1427.208	32266.186	2105.574
197	1427.208	32266.789	2105.574	198	1427.208	32266.186	2105.574	199	1427.208	32266.186	2105.574
200	1427.208	32266.789	2105.574	201	1427.208	32266.186	2105.574	202	1427.208	32266.186	2105.574

202	1555.603	3274.076	1417.963	203	1574.542	3274.314	1419.534	204	1515.467	3295.222	1417.581
205	1544.526	3300.857	1418.280	206	1586.785	3311.748	1419.407	207	1496.310	3315.462	1417.772
208	1505.581	3319.494	1417.645	209	1514.598	3323.844	1419.026	210	1600.521	3320.974	1555.638
211	1658.786	3310.560	1557.049	212	1562.427	3327.727	1554.438	213	1600.521	3324.314	1555.638
214	1523.911	3315.462	1554.678	215	1542.087	3322.844	1555.074	216	1596.778	3322.371	1692.304
217	1639.370	3320.974	1692.629	218	1654.450	3321.037	1693.374	219	1664.016	3320.560	1694.547
220	1606.392	3347.977	1691.732	221	1654.450	3324.215	1693.374	222	1564.040	3320.265	1691.337
223	1587.016	3327.726	1691.302	224	1606.392	3327.076	1691.732	225	1625.370	3327.314	1692.629
226	1569.640	3325.462	1691.337	227	1596.778	3330.857	1692.304	228	1541.536	3311.748	1692.314
229	1550.666	3315.462	1691.838	230	1559.711	3319.494	1691.866	231	1568.572	3323.844	1692.403
232	1649.091	3320.974	1829.506	233	1707.992	3310.560	1831.538	234	1611.816	3327.727	1828.235
235	1649.091	3320.974	1829.506	236	1576.574	3315.462	1829.532	237	1594.354	3323.844	1830.014
238	1644.635	3322.371	1829.506	239	1671.683	3320.974	1831.538	240	1700.276	3321.037	1966.723
241	1730.414	3320.974	1966.723	242	1652.427	3322.844	1965.417	243	1700.276	3324.314	1966.723
244	1617.925	3320.974	1966.723	245	1635.029	3322.844	1965.417	246	1652.427	3327.076	1965.417
247	1671.683	3320.974	1966.723	248	1617.925	3322.844	1965.417	249	1644.635	3330.857	1966.723
250	1593.014	3311.748	1967.188	251	1601.635	3315.462	1965.417	252	1610.542	3319.494	1967.188
253	1619.133	3323.844	1967.188	254	1693.146	3320.974	2102.323	255	1751.580	3320.560	2103.996
256	1657.272	3327.727	2102.323	257	1693.146	3324.314	2102.323	258	1625.370	3315.462	2104.842
259	1643.604	3323.844	2102.323	260	1688.430	3322.371	2241.367	261	1713.480	3320.974	2239.462
262	1741.135	3327.727	2238.827	263	1771.342	3320.560	2239.462	264	1695.272	3327.076	2239.462
265	1741.135	3324.314	2238.827	266	1663.141	3320.266	2241.367	267	1678.492	3327.727	2239.462
268	1695.272	3327.727	2238.827	269	1713.480	3324.314	2239.462	270	1663.141	3329.522	2240.796
271	1688.430	3300.857	2241.367	272	1641.217	3311.748	2243.019	273	1649.218	3315.462	2243.018
274	1657.473	3319.494	2243.018	275	1665.982	3323.844	2244.542	276	1304.540	3136.645	1417.010
277	1326.356	3341.021	1417.102	278	1352.063	3344.661	1417.208	279	1377.279	3147.569	1417.328
280	1402.387	3349.741	1417.462	281	1377.279	3351.181	1417.610	282	1453.892	3151.886	1417.772
283	1440.291	3351.858	1417.948	284	1307.119	3151.096	1418.139	285	1534.378	3149.600	1418.344
286	1552.069	3347.371	1418.562	287	1590.168	3144.408	1418.795	288	1618.739	3140.710	1419.642
289	1294.0126	3369.920	1413.706	290	1345.928	3179.163	1414.534	291	1399.226	3145.132	1415.317
292	1454.019	3369.920	1418.562	293	1510.130	3187.249	1416.756	294	1568.693	3183.395	1417.413
295	1627.374	3376.270	1418.562	296	1524.437	3136.929	1556.936	297	1376.227	3144.897	1556.940
298	1429.151	3347.418	1556.940	299	1483.208	3132.027	1556.974	300	1538.399	3151.190	1556.974
301	1594.723	3341.433	1556.940	302	1652.182	3140.710	1556.936	303	1345.434	3137.097	1655.959
304	1372.723	3341.433	1695.935	305	1406.148	3131.038	1695.935	306	1427.709	3147.906	1695.935
307	1455.407	3350.042	1695.935	308	1483.241	3131.444	1695.935	309	1511.211	3132.112	1695.935
310	1539.318	3352.046	1695.935	311	1567.562	3144.445	1695.935	312	1595.942	3149.713	1695.935
313	1624.458	3347.446	1694.877	314	1653.110	3144.445	1694.877	315	1681.849	3140.710	1694.877
316	1338.859	3370.146	1695.935	317	1395.646	3179.351	1695.935	318	1452.735	3138.282	1695.935
319	1510.125	3337.940	1695.935	320	1567.016	3187.323	1695.935	321	1625.807	3163.434	1694.877
322	1684.101	3376.270	1693.983	323	1367.532	3137.154	1834.078	324	1424.365	3145.085	1834.008
325	1481.155	3350.080	1833.796	326	1537.903	3152.140	1833.443	327	1594.608	3131.265	1832.949
328	1651.271	3347.456	1832.314	329	1537.903	3140.710	1831.538	330	1390.731	3137.097	1971.295
331	1419.883	3350.042	1971.325	332	1448.878	3145.038	1971.304	333	1477.715	3147.907	1971.231
334	1506.395	3350.042	1971.325	335	1534.917	3151.444	1970.930	336	1563.281	3152.112	1970.702
337	1591.468	3352.046	1970.930	338	1619.738	3151.444	1970.930	339	1647.429	3149.713	1969.707
340	1675.164	3347.446	1969.707	341	1702.272	3151.444	1969.707	342	1730.160	3140.710	1969.707
343	1334.446	3347.446	1969.707	344	1444.424	3179.351	1972.449	345	1503.459	3183.282	1972.084
346	1561.602	3347.939	1969.707	347	1618.651	3187.323	1970.608	348	1675.206	3183.282	1969.707
349	1730.868	3347.939	1969.707	350	1415.030	3135.429	2107.312	351	1473.683	3144.897	2107.782
352	1531.127	3349.929	2107.693	353	1587.348	3152.027	2107.312	354	1642.351	3151.190	2106.667
355	1695.135	3347.418	2105.750	356	1748.702	3144.711	2104.560	357	1440.430	3136.646	2243.018
358	1469.338	3341.021	2243.272	359	1498.794	3144.661	2243.441	360	1527.299	3147.569	2243.018
361	1555.351	3349.741	2243.526	362	1562.952	3151.181	2243.441	363	1610.102	3151.886	2243.272
364	1635.800	3351.858	2243.018	365	1563.047	3151.096	2242.579	366	1683.842	3149.600	2242.256
367	1714.186	3371.371	2241.748	368	1739.078	3151.408	2241.156	369	1763.518	3140.710	2240.478
370	1431.032	3369.920	2244.542	371	1492.250	3179.163	2244.747	372	1551.400	3135.132	2244.598
373	1608.451	3369.920	2244.097	374	1692.414	3187.249	2243.244	375	1716.258	3133.396	2242.037
376	1767.074	3376.270	2244.097	377	1284.700	3249.105	1414.883	378	1341.449	3200.634	1418.804
379	1387.662	3331.020	1418.598	380	1280.918	3224.888	1414.072	381	1306.556	3300.349	1417.248
382	1332.417	3309.493	1418.598	383	1358.469	3322.320	1419.026	384	1384.804	3338.830	1418.534
385	1275.489	3343.847	1415.118	386	1325.447	3330.131	1418.677	387	1367.662	3347.022	1418.598
388	1268.472	3345.980	1418.598	389	1291.454	3338.453	1418.121	390	1320.542	3352.546	1419.042
391	1355.435	3345.260	1419.264	392	1346.234	3355.594	1418.783	393	1304.879	3324.883	1555.666
394	1355.435	3349.429	1555.667	395	1408.499	3355.594	1555.719	396	1292.348	3395.980	1555.666
397	1343.106	3352.224	1555.667	398	1419.772	3355.594	1555.665	399	1552.914	3345.105	1555.666
400	1389.757	3350.649	1555.667	401	1436.281	3355.594	1555.665	402	1552.914	3345.105	1555.666
403	1353.325	3350.649	1555.667	404	1378.898	3355.594	1555.665	405	1405.669	3322.682	1691.201
406	1433.603	3350.649	1555.667	407	1322.542	3355.594	1555.665	408	1371.469	3329.893	1692.634
409	1436.261	3350.649	1555.667	410	1313.301	3355.594	1555.665	411	1337.507	3356.294	1694.435
412	1366.262	3350.649	1555.667	413	1402.065	3355.594	1555.665	414	1444.717	3352.816	1691.133
415	1331.022	3350.649	1555.667	416	1402.065	3355.594	1555.665	417	1471.172	3352.816	1826.458
418	1333.830	3350.649	1555.667	419	1393.011	3355.594	1555.665	420	1471.172	3352.816	1826.458
421	1378.209	3350.649	1555.667	422	1437.927	3355.594	1555.665	423	1491.336	3350.496	1964.670
424	1373.209	3350.649	1555.667	425	1395.450	3355.594	1555.665	426	1427.412	3350.496	1964.670
427	1457.072	3350.649	1555.667	428	1488.430	3355.594	1555.665	429	1367.264	3345.647	1971.073
430	1419.554	3350.649	1555.667	431	1491.336	3355.594	1555.665	432	1361.426	3345.647	1971.073
433	1363.799	3350.649	1555.667	434	1414.133	3355.594	1555.665	435	1453.096	3345.647	1971.073
436	1506.630	3350.649	1555.667	437	1394.795	3355.594	1555.665	438	1453.096	3345.647	1971.073
439	1506.630	3350.649	1555.667	440	1394.795	3355.594	1555.665	441	1453.096	3345.647	1971.073
442	1506.630	3350.649	1555.667	443	1394.795	3355.594	1555.665	444	1453.096	3345.647	1971.073
445	1506.630	3350.649	1555.667	446	1394.795	3355.594	1555.665	447	1453.096	3345.647	1971.073
448	1506.630	3350.649	1555.667	449	1394.795	3355.594	1555.665	450	1453.096	3345.647	1971.073
451	1410.553	3350.649	1555.667	452	1469.910	3355.594	1555.665	453	1552.825	3352.816	2242.447
454	1405.378	3350.649	1555.667	455	1430.429	3355.594					

457	1504.334	3347.443	2243.290	458	1561.583	3353.308	2245.304	459	1642.043	3350.375	1418.979
460	1550.438	3323.813	2243.788	461	1657.537	3340.291	1418.471	462	1663.188	3350.392	1418.026
463	1629.659	3324.444	1414.314	464	1639.681	3366.294	1419.149	465	1566.575	3329.173	1419.550
466	1599.005	3303.414	1470.056	467	1502.831	3329.035	1420.058	468	1607.054	3322.038	1419.550
469	1561.225	3320.520	1470.122	470	1520.366	3347.625	1419.149	471	1520.313	3329.145	1419.380
472	1522.215	3333.762	1417.304	473	1520.313	3343.752	1419.210	474	1514.598	3353.054	1418.026
475	1674.174	3324.813	1556.717	476	1684.016	3390.392	1555.328	477	1620.565	3308.435	1556.964
478	1624.498	3352.038	1556.153	479	1549.918	3133.325	1556.253	480	1542.087	3353.055	1555.074
481	1691.520	3325.373	1674.635	482	1697.547	3293.616	1694.322	483	1702.399	3340.291	1693.666
484	1709.775	3339.392	1672.488	485	1672.954	3254.405	1694.133	486	1682.547	3308.294	1692.752
487	1639.926	3329.038	1674.836	488	1644.568	3308.456	1693.849	489	1644.611	3329.067	1693.512
490	1691.956	3352.038	1672.826	491	1612.489	3120.590	1693.479	492	1613.997	3347.625	1692.710
493	1574.704	3329.210	1692.847	494	1576.715	3339.667	1692.795	495	1574.704	3343.815	1692.847
496	1566.672	3353.054	1692.403	497	1720.809	3253.618	1721.602	498	1720.466	3340.392	1829.506
499	1644.014	3308.476	1830.712	500	1674.427	3352.038	1829.570	501	1602.609	3345.909	1830.014
502	1694.354	3353.054	1830.014	503	1737.864	3250.375	1828.628	504	1743.960	3323.619	1968.557
505	1744.701	3340.291	1827.804	506	1732.088	3390.393	1864.484	507	1718.712	3343.487	1967.521
508	1722.327	3366.294	1968.194	509	1683.705	3290.237	1967.855	510	1690.902	3338.497	1967.555
511	1697.305	3329.093	1957.384	512	1636.914	3352.038	1966.384	513	1660.531	3330.653	1967.340
514	1660.849	3324.476	1968.926	515	1625.463	3324.273	1967.464	516	1627.600	3335.952	1967.315
517	1622.463	3345.679	1967.464	518	1619.133	3353.055	1967.903	519	1766.799	3323.618	2109.188
520	1776.642	3390.393	2103.121	521	1713.233	3306.559	2104.375	522	1719.414	3352.038	2103.269
523	1651.087	3339.994	2104.899	524	1643.069	3353.055	2105.984	525	1781.076	3350.375	2240.955
526	1789.426	3329.318	2241.494	527	1756.443	3340.291	2241.082	528	1802.126	3330.392	2239.716
529	1762.931	3352.038	2241.097	530	1773.996	3366.294	2243.404	531	1727.212	3320.269	2241.605
532	1733.006	3330.392	2241.177	533	1739.912	3329.130	2240.714	534	1741.928	3352.038	2240.224
535	1709.654	3330.392	2241.732	536	1702.924	3347.625	2241.907	537	1672.650	3329.337	2243.209
538	1674.872	3330.392	2242.704	539	1672.650	3343.942	2243.209	540	1665.952	3353.054	2244.542
541	1321.304	2974.554	1418.026	542	1342.256	2975.519	1417.737	543	1363.655	2976.260	1417.547
544	1389.534	2976.817	1417.455	545	1497.862	2977.191	1417.442	546	1430.648	2977.381	1417.568
547	1453.892	2977.388	1417.772	548	1477.595	2977.212	1418.075	549	1501.757	2976.852	1418.478
550	1526.377	2976.369	1418.979	551	1551.457	2975.582	1419.578	552	1570.494	2974.672	1420.277
553	1602.990	2977.578	1421.074	554	1511.398	3064.002	1417.772	555	1357.230	3068.006	1417.753
556	1409.660	3070.407	1417.857	557	1454.591	3071.208	1417.942	558	1504.720	3070.408	1418.111
559	1556.050	3090.393	1418.301	560	1603.579	3064.002	1418.534	561	1344.305	2974.594	1553.176
562	1388.396	2976.260	1553.157	563	1434.892	2977.191	1553.530	564	1481.691	2977.389	1554.126
565	1529.855	2976.852	1555.064	566	1579.302	2975.982	1556.254	567	1629.914	2973.579	1557.726
568	1367.589	2974.554	1603.185	569	1396.763	2975.519	1603.345	570	1414.060	2976.259	1609.628
571	1437.541	2976.317	1603.912	572	1461.145	2977.191	1603.219	573	1484.893	2977.381	1690.608
574	1509.784	2977.383	1691.019	575	1532.819	2977.212	1691.474	576	1556.497	2976.552	1691.970
577	1581.319	2976.309	1692.508	578	1605.764	2975.582	1693.699	579	1630.323	2974.672	1693.713
580	1659.145	2973.378	1694.378	581	1358.614	3064.115	1690.279	582	1408.854	3067.732	1690.763
583	1459.183	3059.795	1691.301	584	1509.617	3070.304	1691.838	585	1560.139	3069.258	1692.374
586	1610.756	3086.059	1692.911	587	1661.467	3062.505	1693.447	588	1391.154	2974.594	1826.204
589	1439.217	2976.259	1826.867	590	1487.222	2977.191	1827.537	591	1535.172	2977.388	1828.363
592	1583.085	2975.852	1829.196	593	1630.962	2975.582	1830.065	594	1678.682	2973.578	1831.030
595	1419.002	2974.594	1834.183	596	1439.740	2975.519	1844.552	597	1464.306	2974.260	1844.907
598	1489.701	2975.317	1953.248	599	1512.923	2977.191	1965.575	600	1536.975	2977.381	1968.888
601	1569.854	2977.317	1968.136	602	1564.562	2977.212	1966.471	603	1603.098	2976.852	1969.741
604	1633.463	2976.369	1968.498	605	1634.556	2975.982	1967.260	606	1677.677	2974.672	1967.468
607	1730.326	2977.379	1968.583	608	1404.566	2975.582	1967.260	609	1457.775	3067.715	1969.736
610	1509.950	3069.607	1968.327	611	1551.115	3070.092	1968.914	612	1611.268	3069.071	1967.259
613	1586.411	3066.342	1967.661	614	1706.941	3062.505	1969.621	615	1439.132	2974.594	2103.121
616	1439.344	2976.260	2103.746	617	1538.248	2977.191	2104.293	618	1585.831	2977.388	2104.489
619	1632.097	2976.852	2104.607	620	1677.045	2975.582	2104.555	621	1720.677	2973.579	2104.334
622	1463.544	2974.594	2243.018	623	1489.188	2975.519	2243.237	624	1514.344	2975.260	2243.389
625	1539.014	2976.317	2243.452	626	1563.197	2977.191	2243.470	627	1586.893	2977.381	2243.406
628	1610.102	2977.386	2243.272	629	1632.824	2977.212	2243.067	630	1655.030	2976.852	2242.792
631	1676.809	2975.369	2243.447	632	1693.071	2975.582	2242.031	633	1718.648	2974.572	2241.544
634	1739.134	2973.378	2243.886	635	1449.320	3064.002	2242.744	636	1504.692	3067.653	2242.891
637	1597.947	3059.843	2242.533	638	1609.086	3070.573	2242.801	639	1658.108	3067.843	2242.754
640	1709.013	3067.653	2242.553	641	1745.802	3064.002	2242.254	642	1381.623	3384.423	1418.026
643	1362.452	3413.252	1418.026	644	1312.795	3404.016	1416.502	645	1386.963	3413.252	1556.313
646	1432.683	3384.423	1692.897	647	1411.474	3413.252	1692.600	648	1360.589	3464.532	1692.008
649	1435.983	3413.252	1829.357	650	1483.757	3364.423	1967.767	651	1460.496	3413.252	1967.174
652	1408.384	3404.447	1957.513	653	1483.007	3413.252	2104.461	654	1534.791	3384.423	2242.637
655	1509.516	3413.252	2241.748	656	1450.178	3404.362	2243.918	657	1540.125	3384.804	1418.534
658	1569.652	3416.554	1419.042	659	1614.420	3403.473	1418.514	660	1589.232	3416.554	1556.922
661	1588.936	3403.805	1694.167	662	1612.812	3416.554	1694.802	663	1661.156	3403.473	1693.362
664	1636.391	3416.554	1932.631	665	1637.746	3384.604	1969.799	666	1659.971	3416.554	1970.261
667	1707.892	3403.473	1978.190	668	1583.550	3416.554	2243.018	669	1685.956	3384.604	2245.431
670	1707.130	3416.554	2246.320	671	1754.828	3403.473	2243.823	672	1469.823	3363.214	1419.123
673	1419.840	3370.393	1419.042	674	1415.792	3380.486	1418.651	675	1387.548	3401.695	1418.534
676	1412.744	3390.393	1556.118	677	1458.153	3370.834	1556.118	678	1439.711	3390.138	1556.795
679	1463.829	3363.214	1693.574	680	1473.365	3370.834	1693.956	681	1470.621	3360.487	1694.251
682	1433.076	3401.696	1832.300	683	1466.677	3390.138	1832.300	684	1500.628	3370.834	1831.792
685	1473.643	3390.138	1832.300	686	1517.435	3363.214	1969.995	687	1527.691	3370.834	1967.630
688	1524.290	3386.487	1969.995	689	1476.223	3401.695	1969.614	690	1520.609	3390.138	1970.053
691	1533.153	3370.393	2243.467	692	1547.576	3380.486	2243.467	693	1571.240	3363.214	2244.415
694	1582.416	3370.834	2245.431	695	1575.479	3380.486	2245.431	696	1542.030	3401.695	2243.653
697	1574.542	3390.138	2245.556	698	1501.390	3360.486	1419.550	699	1526.050	3402.833	1419.042
700	1493.775	3326.776	1419.550	701	1490.818	3378.454	1419.550	702	1497.860	3360.122	1419.550
703	1521.033	3356.040	1527.091	704	1526.743	3369.122	1554.954	705	1555.069	3360.474	1694.532
706	1575.963	3402.358	1693.786	707	1548.295	3368.294	1694.632	708	1550.963	3378.708	1694.565
709	1593.050	3384.122	1834.278	710	1575.558	3368.548	1832.173	711	1580.511	3385.122	1831.792

712	1608	747	3360	9228	1969	714	1629	745	3402	838	1948	529	714	1602	821	3368	802	1969	714
715	1605	107	3378	902	1969	714	1607	743	3389	122	1969	706	717	1630	083	3369	056	2107	255
716	1634	274	3389	122	2106	620	1602	425	3361	182	2244	795	720	1680	587	3402	838	2243	272
721	1657	345	3389	122	2244	796	1659	221	3377	216	2244	415	723	1661	156	3369	122	2244	034
724	1256	467	3389	122	1417	718	1337	350	3455	924	1419	550	726	1250	438	3465	896	1417	010
727	1299	879	3389	122	1418	334	1347	720	3455	596	1420	058	729	1274	145	3485	896	1555	229
730	1371	365	3389	122	1557	345	1303	490	3440	333	1593	770	732	1402	754	3455	924	1694	082
733	1297	851	3389	122	1593	447	1346	450	3440	240	1604	040	735	1395	649	3498	576	1694	632
736	1321	558	3389	122	1831	665	1418	713	3440	596	1831	919	738	1351	573	3440	938	1970	222
739	1456	167	3389	122	1968	614	1345	265	3440	896	1968	694	741	1303	821	3492	245	1969	535
742	1442	377	3389	122	1969	206	1368	972	3440	896	2108	102	744	1460	042	3498	596	2106	493
745	1399	155	3389	122	2246	574	1497	500	3455	924	2243	145	747	1392	678	3498	896	2246	320
748	1441	192	3389	122	2245	050	1459	706	3498	596	2243	783	750	1573	525	3458	591	1423	106
751	1667	252	3389	122	1420	439	1582	670	3500	528	1422	344	753	1626	136	3488	690	1421	201
754	1673	602	3389	122	1420	058	1605	753	3500	528	1559	033	756	1696	621	3476	752	1557	388
757	1626	474	3389	122	1420	336	1715	336	3433	573	1604	374	759	1629	237	3500	628	1695	723
760	1676	439	3389	122	1695	225	1723	640	3476	752	1604	717	762	1652	520	3500	628	1832	427
763	1748	559	3389	122	1832	046	1657	422	3458	591	1969	630	765	1764	619	3433	573	1968	317
766	1675	804	3389	122	1969	122	1724	741	3458	690	1969	249	768	1773	678	3476	752	1969	375
769	1699	087	3389	122	2105	816	1798	097	3476	752	2106	705	771	1714	369	3458	591	2242	891
772	1813	302	3389	122	2242	256	1722	370	3500	628	2242	513	774	1773	043	3488	690	2243	272
775	1823	716	3389	122	2244	034	1726	724	3426	926	1413	796	777	1796	466	3449	320	1419	550
778	1392	043	3389	122	1419	677	1367	532	3483	463	1419	423	780	1387	598	3471	926	1419	804
781	1421	343	3389	122	1555	372	1412	913	3471	926	1555	753	783	1429	212	3423	676	1693	024
784	1447	203	3389	122	1693	701	1442	716	3460	623	1603	447	786	1416	594	3483	737	1693	828
787	1438	224	3389	122	1968	233	1472	221	3449	320	1830	014	789	1463	544	3471	926	1830	649
790	1478	859	3389	122	1968	233	1497	513	3449	320	1966	844	792	1493	389	3460	623	1967	217
793	1465	576	3389	122	2104	246	1438	380	3471	926	1947	538	795	1523	276	3449	320	2103	657
796	1514	175	3389	122	2240	986	1528	187	3434	245	2242	637	798	1548	634	3471	926	2240	478
799	1544	062	3389	122	1418	407	1514	598	3434	245	1417	518	801	1539	490	3445	018	2241	474
802	1541	141	3389	122	1417	264	1553	387	3471	418	1416	754	804	1515	614	3445	018	1517	772
805	1520	287	3389	122	1692	346	1525	520	3431	286	1692	585	807	1541	268	3446	018	1555	059
808	1551	301	3389	122	1692	346	1553	387	3458	719	1692	585	810	1602	735	3460	023	1691	923
811	1555	922	3389	122	1692	346	1572	002	3471	418	1691	796	813	1577	082	3471	418	1691	246
814	1542	776	3389	122	1968	233	1602	863	3471	418	1829	490	816	1637	830	3431	286	1965	583
817	1551	855	3389	122	1968	233	1618	230	3446	018	1966	920	819	1623	437	3458	713	1966	328
820	1623	644	3389	122	1968	233	1643	884	3446	018	2104	707	822	1654	425	3471	418	2102	979
823	1680	175	3389	122	2241	240	1701	034	3486	023	2240	732	825	1669	538	3446	018	2241	494
826	1674	872	3389	122	2240	959	1686	206	3471	418	2240	724	828	1644	088	3531	743	1415	486
829	1339	211	3389	122	1421	201	1237	738	3577	590	1413	962	831	1284	220	3564	448	1418	153
832	1330	702	3389	122	1422	344	1261	021	3544	951	1552	223	834	1353	647	3591	306	1558	827
835	1291	679	3389	122	1691	983	1385	020	3544	951	1604	971	837	1284	905	3577	590	1590	464
838	1330	448	3389	122	1692	397	1376	092	3531	306	1695	310	840	1307	588	3577	590	1828	744
841	1399	536	3389	122	1831	792	1338	068	3531	743	1968	444	843	1432	429	3544	951	1968	741
844	1330	871	3389	122	1967	007	1376	076	3594	448	1967	440	846	1422	481	3591	306	1968	275
847	1394	155	3389	122	2105	266	1445	425	3591	306	2104	758	849	1385	053	3531	743	2244	423
850	1479	038	3389	122	2242	510	1377	438	3577	590	2243	576	852	1422	904	3584	448	2242	383
853	1465	370	3389	122	2241	240	1592	068	3545	713	1420	593	855	1681	934	3527	044	1419	169
856	1599	342	3389	122	1420	820	1644	646	3584	067	1419	550	858	1689	350	3577	336	1419	280
859	1623	141	3389	122	1557	684	1713	819	3577	336	1555	864	861	1638	127	3545	713	1594	632
862	1731	345	3389	122	1694	124	1640	340	3590	798	1604	548	864	1692	314	3584	067	1693	997
865	1738	287	3389	122	1968	444	1669	538	3530	798	1831	411	867	1762	756	3577	336	1831	030
868	1644	185	3389	122	1968	444	1780	706	3527	044	1969	079	870	1692	737	3590	798	1968	275
871	1739	981	3389	122	2106	197	1787	225	3577	336	1968	614	873	1715	935	3590	798	2105	138
874	1511	093	3389	122	2242	002	1730	244	3545	713	2242	510	876	1830	086	3521	044	2244	034
877	1739	134	3389	122	1421	074	1787	648	3584	067	2242	391	879	1836	162	3577	336	2243	780
880	1362	660	3389	122	1420	058	1378	494	3533	394	1421	074	882	1363	025	3564	763	1420	439
883	1359	150	3389	122	1550	753	1387	958	3594	133	1419	804	885	1403	007	3533	394	1557	811
886	1412	913	3389	122	1694	124	1411	220	3533	394	1604	251	888	1427	561	3533	394	1694	548
889	1432	695	3389	122	1831	284	1407	410	3533	394	1604	505	891	1438	229	3594	132	1693	701
892	1452	114	3389	122	1968	021	1463	764	3533	394	1831	649	894	1459	480	3545	403	1968	444
895	1476	067	3389	122	2242	637	1482	764	3533	394	2104	758	897	1455	670	3545	403	1967	636
898	1488	740	3389	122	2241	387	1501	274	3533	394	2241	494	900	1514	175	3545	403	2104	546
901	1507	940	3389	122	1418	661	1525	774	3533	394	2241	494	903	1532	632	3564	763	2241	494
904	1503	930	3389	122	1418	661	1539	490	3529	076	1417	010	906	1559	683	3514	352	1417	645
907	1562	604	3389	122	1692	384	1537	712	3529	076	1554	594	909	1531	615	3560	064	1416	883
910	1522	220	3389	122	1692	384	1562	647	3529	076	1554	594	912	1551	005	3591	352	1554	277
913	1607	985	3389	122	1831	284	1511	330	3529	076	1554	594	915	1587	581	3529	076	1692	177
916	1582	035	3389	122	1967	007	1576	490	3514	852	1601	833	918	1512	515	3529	076	1629	760
919	1601	974	3389	122	2106	197	1656	208	3514	852	1947	132	921	1620	056	3590	798	1966	963
922	1637	449	3389	122	2241	113	1632	454	3560	064	2242	391	924	1627	459	3591	352	1966	920</

967	1309.677	3187.694	1191.557	968	1339.564	3191.059	1191.684	969	1369.310	3193.733	1191.829
970	1398.915	3195.716	1192.023	971	1428.379	3197.007	1192.224	972	1457.702	3197.606	1192.474
973	1486.884	3197.515	1192.742	974	1515.925	3196.732	1193.039	975	1544.824	3195.257	1193.363
976	1573.583	3193.091	1193.716	977	1602.200	3190.234	1194.097	978	1630.076	3186.684	1194.506
979	1289.045	3155.442	1036.645	980	1345.999	3163.034	1036.794	981	1402.217	3187.360	1036.956
982	1457.702	3164.920	1037.153	983	1512.453	3169.215	1037.379	984	1565.471	3165.744	1037.633
985	1519.754	3159.506	1037.915	986	1278.378	3183.636	1036.645	987	1338.859	3191.659	1036.645
989	1398.633	3199.715	1038.339	989	1457.702	3197.606	1037.153	990	1516.066	3196.731	1037.661
991	1373.724	3193.041	1038.339	992	1630.076	3185.634	1037.153	993	1289.046	3155.442	831.324
994	1317.614	3159.584	831.391	995	1345.999	3163.034	831.445	996	1374.200	3165.793	831.546
997	1402.217	3167.860	831.655	998	1436.051	3169.215	831.730	999	1457.702	3169.920	831.832
1000	1485.170	3169.913	831.941	1001	1512.453	3169.214	832.059	1002	1539.554	3167.825	832.181
1003	1566.471	3165.743	832.312	1004	1593.204	3162.970	832.450	1005	1619.754	3159.506	832.594
1006	1285.236	3169.920	831.324	1007	1343.635	3177.393	831.416	1008	1401.314	3183.100	831.606
1009	1458.274	3185.541	831.855	1010	1514.314	3185.216	832.293	1011	1570.933	3182.126	832.770
1012	1624.834	3176.270	833.356	1013	1278.378	3183.636	831.324	1014	1308.707	3137.693	831.303
1015	1338.859	3191.059	831.324	1016	1368.834	3193.733	831.388	1017	1398.633	3195.715	831.493
1018	1426.256	3197.006	831.642	1019	1457.702	3197.606	831.832	1020	1486.972	3197.514	832.065
1021	1517.066	3196.731	832.340	1022	1544.983	3195.257	832.659	1023	1573.724	3193.091	833.018
1024	1502.288	3190.233	833.420	1025	1630.076	3186.684	833.864	1026	1289.046	3155.442	726.603
1027	1345.999	3163.034	726.144	1028	1402.217	3157.860	726.314	1029	1457.702	3169.920	726.511
1030	1512.453	3169.920	726.727	1031	1566.471	3165.744	726.991	1032	1619.754	3159.506	727.273
1033	1278.378	3183.636	726.003	1034	1338.859	3191.059	726.003	1035	1398.633	3195.715	726.172
1036	1457.702	3197.606	726.511	1037	1516.066	3196.731	727.019	1038	1573.724	3193.091	727.697
1039	1630.076	3186.684	728.343	1040	1289.046	3155.442	570.682	1041	1317.614	3159.506	570.749
1042	1345.999	3163.034	570.323	1043	1374.200	3165.793	570.904	1044	1402.217	3167.660	570.992
1045	1430.052	3169.920	571.088	1046	1457.702	3169.920	571.100	1047	1485.170	3169.913	571.299
1048	1512.453	3169.920	571.416	1049	1539.554	3167.825	571.539	1050	1565.471	3165.744	571.670
1051	1593.204	3162.971	571.867	1052	1619.754	3159.506	571.952	1053	1285.236	3169.920	570.682
1054	1343.635	3177.393	570.774	1055	1401.314	3183.100	570.964	1056	1458.274	3185.541	571.253
1057	1514.314	3185.217	571.641	1058	1570.933	3182.126	572.129	1059	1624.834	3176.270	572.714
1060	1278.378	3183.636	570.662	1061	1308.707	3193.733	570.661	1062	1338.859	3191.059	570.582
1063	1368.834	3193.733	570.745	1064	1398.633	3195.716	570.861	1065	1428.256	3197.007	571.080
1066	1457.702	3197.606	571.190	1067	1486.972	3197.606	571.423	1068	1516.066	3196.732	571.698
1069	1544.983	3199.727	572.016	1070	1573.724	3193.091	572.375	1071	1602.289	3190.234	572.778
1072	1630.076	3186.684	573.222	1073	1284.274	3145.552	447.746	1074	1342.182	3153.647	447.026
1075	1399.310	3153.772	446.545	1076	1426.061	3161.030	446.603	1077	1511.333	3160.720	446.906
1078	1567.225	3157.841	447.535	1079	1622.040	3152.394	448.508	1080	1282.696	3195.574	454.096
1081	1341.490	3203.611	453.781	1082	1400.100	3208.754	453.390	1083	1458.527	3211.005	453.524
1084	1516.771	3210.363	452.483	1085	1574.831	3206.828	454.765	1086	1632.709	3200.400	455.874
1087	1285.030	3137.154	327.350	1088	1318.390	3140.626	327.262	1089	1348.172	3143.504	327.269
1090	1377.375	3145.790	327.191	1091	1409.949	3147.484	327.209	1092	1434.045	3148.584	327.252
1093	1461.512	3144.092	327.350	1094	1486.972	3149.008	327.474	1095	1514.711	3148.330	327.632
1096	1546.443	3147.060	327.826	1097	1565.596	3145.198	328.056	1098	1590.170	3142.742	328.320
1099	1614.166	3139.694	328.620	1100	1282.696	3169.920	328.620	1101	1344.228	3177.717	328.472
1102	1403.585	3182.817	328.451	1103	1451.068	3185.223	328.557	1104	1516.375	3184.934	328.789
1105	1569.610	3181.950	329.149	1106	1620.170	3176.270	329.634	1107	1277.870	3206.242	327.350
1108	1309.539	3210.797	327.364	1109	1340.742	3214.596	327.005	1110	1371.301	3217.640	327.477
1111	1401.794	3217.930	327.576	1112	1431.628	3221.465	327.703	1113	1461.004	3222.244	327.858
1114	1489.921	3222.269	328.041	1115	1518.380	3221.539	328.253	1116	1548.380	3220.053	328.493
1117	1573.922	3217.813	328.761	1118	1601.004	3214.318	329.058	1119	1627.628	3211.068	329.382

SET UP CO-EFFICIENT MATRICES

NODAL POINT LINKAGES

ELMT NO ELMT TYPE NODAL POINTS

1	HX20	1	2	3	6	7	9	10	11	22	23
2	HX20	25	26	31	32	33	36	37	39	40	41
3	HX20	3	4	5	7	8	11	12	13	23	24
4	HX20	26	27	33	34	35	37	38	41	42	43
5	HX20	9	10	11	14	15	17	18	19	25	26
6	HX20	28	29	39	40	41	44	45	47	48	49
7	HX20	11	12	13	15	16	19	20	21	26	27
8	HX20	23	30	41	42	43	45	46	49	50	51
9	HX20	31	32	33	36	37	39	40	41	52	53
10	HX20	35	36	61	62	63	66	67	69	70	71
11	HX20	33	34	35	37	38	41	42	43	72	73
12	HX20	25	27	63	64	65	67	68	71	72	73
13	HX20	39	40	41	44	45	47	48	49	55	56
14	HX20	58	59	69	70	71	74	75	77	78	79
15	HX20	41	42	43	45	46	49	50	51	56	57
16	HX20	59	60	71	72	73	75	76	79	80	81
17	HX20	61	62	63	66	67	69	70	71	82	83
18	HX20	85	86	91	92	93	96	97	99	100	101
19	HX20	63	64	65	67	68	71	72	73	83	84
20	HX20	86	87	93	94	95	97	98	101	102	103
21	HX20	69	70	71	74	75	77	78	79	85	86
22	HX20	88	89	99	100	101	104	105	107	108	109
23	HX20	71	72	73	75	76	79	80	81	86	87
24	HX20	89	90	101	102	103	105	106	109	110	111
25	HX20	112	113	114	116	117	118	119	120	128	129
26	HX20	130	131	134	135	136	138	139	140	141	142
27	HX20	114	115	1	117	6	120	121	9	129	22
28	HX20	131	132	136	137	31	139	36	142	143	39
29	HX20	118	119	120	122	123	124	125	126	130	131
30	HX20	132	133	140	141	142	144	145	146	147	148
31	HX20	120	121	9	123	14	126	127	17	131	25
32	HX20	133	134	142	143	39	145	44	148	149	47
33	HX20	134	135	136	138	139	140	141	142	150	151
34	HX20	136	137	156	157	158	160	161	162	163	164
35	HX20	153	154	31	139	36	142	143	39	151	52
36	HX20	140	141	142	144	145	146	147	148	152	153
37	HX20	154	155	162	163	164	166	167	168	169	170
38	HX20	142	143	39	145	44	148	149	47	153	55
39	HX20	155	156	164	165	69	167	74	170	171	77
40	HX20	158	159	158	160	151	162	163	164	172	173
41	HX20	174	175	176	179	180	182	183	184	185	186
42	HX20	158	159	61	161	66	164	165	69	173	82
43	HX20	175	185	180	181	91	183	96	186	187	99
44	HX20	162	163	164	166	167	168	169	170	174	175
45	HX20	176	177	184	185	186	188	189	190	191	192
46	HX20	184	185	69	167	74	170	171	77	175	85
47	HX20	177	186	186	187	99	189	104	192	193	107
48	HX20	5	194	195	8	198	13	200	201	24	210
49	HX20	193	196	197	198	217	38	220	43	222	223
50	HX20	212	213	217	218	219	201	202	203	210	211
51	HX20	13	200	201	16	234	21	206	207	27	212
52	HX20	30	214	43	222	223	46	226	51	228	229
53	HX20	201	202	203	204	205	207	208	209	212	213
54	HX20	214	215	223	224	225	226	227	229	230	231
55	HX20	35	216	217	38	220	43	222	223	54	232
56	HX20	57	234	67	238	237	68	242	73	244	245
57	HX20	217	218	219	220	221	223	224	225	232	233
58	HX20	234	235	239	240	241	242	243	245	246	247
59	HX20	43	222	223	40	246	51	248	229	57	234
60	HX20	60	236	73	244	245	76	248	81	250	251
61	HX20	32	223	224	225	226	227	229	230	231	234

33	HX20	235	237	245	246	247	248	249	251	252	253
34	HX20	65	236	239	268	242	73	244	245	84	254
35	HX20	87	256	95	260	261	98	264	103	266	267
36	HX20	239	240	241	242	243	245	246	247	254	255
37	HX20	256	257	261	262	263	264	265	267	268	269
38	HX20	73	244	245	76	248	81	250	251	87	256
39	HX20	95	256	103	268	267	106	270	111	272	273
40	HX20	245	246	247	248	249	251	252	253	256	257
41	HX20	258	259	267	268	269	270	271	273	274	275
42	HX20	276	277	278	289	290	112	113	114	276	297
43	HX20	126	129	303	304	305	316	317	134	135	136
44	HX20	273	279	280	290	291	114	115	1	297	299
45	HX20	127	22	305	306	307	317	318	136	137	31
46	HX20	280	281	282	291	292	1	2	3	253	299
47	HX20	22	23	307	308	309	318	319	31	32	33
48	HX20	282	283	284	292	293	311	312	3	299	300
49	HX20	23	24	309	310	311	319	320	33	34	35
50	HX20	284	285	286	293	294	313	326	144	145	300
51	HX20	24	210	311	312	313	326	321	145	146	217
52	HX20	286	287	288	294	295	315	321	146	147	218
53	HX20	210	211	313	314	315	321	322	217	218	219
54	HX20	303	304	305	316	317	317	318	156	157	323
55	HX20	150	151	330	331	332	343	344	156	157	158
56	HX20	305	306	307	317	318	318	319	31	32	33
57	HX20	151	52	332	333	334	344	345	158	159	61
58	HX20	307	308	309	318	319	319	320	33	34	35
59	HX20	52	53	334	335	336	345	346	61	62	63
60	HX20	309	310	311	319	320	320	321	33	34	35
61	HX20	53	54	336	337	338	346	347	63	64	65
62	HX20	311	312	313	320	321	321	322	217	218	219
63	HX20	54	55	338	339	340	347	348	65	66	67
64	HX20	313	314	315	321	322	322	323	218	219	220
65	HX20	282	283	340	341	342	348	349	219	220	221
66	HX20	330	331	332	343	344	344	345	158	159	160
67	HX20	172	173	357	358	359	370	371	178	179	180
68	HX20	332	333	334	344	345	345	346	61	62	63
69	HX20	173	82	359	360	361	371	372	180	181	91
70	HX20	334	335	336	345	346	346	347	62	63	64
71	HX20	82	83	361	362	363	372	373	91	92	93
72	HX20	336	337	338	346	347	347	348	63	64	65
73	HX20	83	84	363	364	365	373	374	93	94	95
74	HX20	338	339	340	347	348	348	349	239	240	241
75	HX20	84	85	365	366	367	374	375	95	96	97
76	HX20	340	341	342	348	349	349	350	241	242	243
77	HX20	254	255	367	368	369	375	376	201	202	203
78	HX20	112	116	119	377	378	380	381	382	128	130
79	HX20	393	394	134	138	140	399	400	402	403	404
80	HX20	118	122	124	378	379	382	383	384	130	132
81	HX20	394	395	140	144	146	400	401	404	405	406
82	HX20	380	381	382	385	386	388	389	390	393	394
83	HX20	396	397	402	403	404	407	408	410	411	412
84	HX20	382	383	384	386	387	390	391	392	394	395
85	HX20	397	398	404	405	406	408	409	412	413	414
86	HX20	134	136	140	399	400	402	403	404	150	152
87	HX20	415	416	156	160	162	421	422	424	425	426
88	HX20	140	144	146	400	401	404	405	406	152	154
89	HX20	416	417	162	166	168	422	423	426	427	428
90	HX20	402	403	404	407	408	410	411	412	415	416
91	HX20	418	419	424	425	426	429	430	432	433	434
92	HX20	404	405	406	408	409	412	413	414	416	417
93	HX20	419	420	426	427	428	430	431	434	435	436
94	HX20	155	160	162	421	422	424	425	426	172	174
95	HX20	437	438	176	182	184	443	444	446	447	448
96	HX20	162	166	168	422	423	426	427	428	174	175
97	HX20	438	439	184	188	190	444	445	448	449	450
98	HX20	424	425	426	429	430	432	433	434	437	438
99	HX20	440	441	446	447	448	451	452	454	455	456
00	HX20	420	427	428	430	431	434	435	436	438	439
01	HX20	441	442	448	449	450	452	453	456	457	458
02	HX20	197	459	460	199	463	203	465	466	211	47
03	HX20	213	477	219	481	482	221	485	225	487	48
04	HX20	450	461	462	463	464	466	467	468	475	49
05	HX20	477	478	482	483	484	489	490	498	499	50
06	HX20	203	465	466	205	459	471	472	213	477	49
07	HX20	215	474	225	437	458	227	491	231	493	49
08	HX20	466	467	468	469	470	472	473	474	477	47
09	HX20	479	480	488	489	490	491	492	494	495	49
10	HX20	219	461	482	221	485	225	487	488	233	49
11	HX20	235	494	241	503	504	243	507	247	509	51
12	HX20	702	493	494	495	496	498	499	490	497	49
13	HX20	499	500	504	505	506	507	508	510	511	51
14	HX20	225	487	488	227	491	231	493	494	235	49
15	HX20	237	501	247	509	510	249	511	253	255	49
16	HX20	488	489	490	491	492	494	495	496	499	50
17	HX20	501	502	510	511	512	513	514	516	517	50

FIXED DISPLACEMENT

NODAL POINT	TYPE	NODAL POINT	TYPE	NODAL POINT	TYPE	NODAL POINT	TYPE	NODAL POINT	TYPE
112	1	112	3	128	1	129	3	134	1
134	3	150	1	150	3	156	1	156	3
172	1	172	3	178	1	178	3	179	1
197	3	211	1	211	3	219	1	219	3
233	1	233	3	241	1	241	3	255	1
255	3	263	1	263	3	276	1	276	3
268	1	268	3	290	1	296	3	302	1
302	3	303	1	303	3	315	1	315	3
323	1	323	3	329	1	329	3	330	1
330	3	342	1	342	3	350	1	350	3
356	1	356	3	357	1	357	3	369	1
359	3	377	1	377	3	380	1	380	3
385	1	385	3	386	1	388	3	393	1
393	3	396	1	396	3	399	1	399	3
402	1	402	3	407	1	407	3	410	1
410	3	415	1	415	3	418	1	418	3
421	1	421	3	424	1	424	3	429	1
429	3	432	1	432	3	437	1	437	3
440	1	440	3	443	1	443	3	446	1
446	3	451	1	451	3	454	1	454	3
459	1	459	3	460	1	460	3	461	1
461	3	462	1	462	3	475	1	475	3
476	1	476	3	481	1	481	3	482	1
482	3	483	1	483	3	484	1	484	3
497	1	497	3	498	1	498	3	503	1
503	3	504	1	504	3	505	1	505	3
506	1	506	3	519	1	519	3	520	1
520	3	525	1	525	3	526	1	526	3
527	1	527	3	528	1	528	3	541	1
541	3	553	1	553	3	554	1	554	3
560	1	560	3	561	1	561	3	567	1
567	3	568	1	568	3	580	1	580	3
581	1	581	3	587	1	587	3	588	1
588	3	594	1	594	3	595	1	595	3
607	1	607	3	608	1	608	3	614	1
614	3	615	1	615	3	621	1	621	3
622	1	622	3	634	1	634	3	635	1
635	3	641	1	641	3	724	1	724	3
726	1	726	3	729	1	729	3	731	1
731	3	733	1	733	3	736	1	736	3
736	1	738	3	740	1	740	3	743	1
743	3	745	1	745	3	747	1	747	3
751	1	751	3	754	1	754	3	756	1
756	3	758	1	758	3	761	1	761	3
763	1	763	3	765	1	765	3	768	1
768	3	770	1	770	3	772	1	772	3
775	1	775	3	828	1	828	3	830	1
830	3	833	1	833	3	835	1	835	3
837	1	837	3	840	1	840	3	842	1
842	3	844	1	844	3	847	1	847	3
849	1	849	3	851	1	851	3	855	1
855	3	858	1	858	3	860	1	860	3
862	1	862	3	865	1	865	3	867	1
867	3	869	1	869	3	872	1	872	3
874	1	874	3	876	1	876	3	877	1
879	3	1087	1	1087	2	1087	3	1088	1
1088	2	1088	2	1089	1	1089	2	1089	3
1090	1	1090	2	1090	3	1091	1	1091	2
1091	3	1092	1	1092	2	1092	3	1093	1
1093	2	1093	3	1094	1	1094	2	1094	3
1095	1	1095	2	1095	3	1096	1	1096	2
1096	3	1097	1	1097	2	1097	3	1098	1
1098	2	1098	3	1099	1	1099	2	1099	3
1100	1	1100	2	1100	3	1101	1	1101	2
1101	3	1102	1	1102	2	1102	3	1103	1
1103	2	1103	3	1104	1	1104	2	1104	3
1105	1	1105	2	1105	3	1106	1	1106	2
1106	3	1107	1	1107	2	1107	3	1108	1
1108	2	1108	3	1109	1	1109	2	1109	3
1110	1	1110	2	1110	3	1111	1	1111	2
1111	3	1112	1	1112	2	1112	3	1113	1
1113	2	1113	3	1114	1	1114	2	1114	3
1115	1	1115	2	1115	3	1116	1	1116	2
1116	3	1117	1	1117	2	1117	3	1118	1
1118	2	1118	3	1119	1	1119	2	1119	3