4

IMPLEMENTING A COMPUTER-BASED ANALYSIS PROCEDURE

This chapter contains material previously published in Shaw (1992), by kind permission of Prentice Hall.

In Chapters 1–3, the design process, the evaluation of designs and the use of computers in evaluation were considered, together with the techniques for analysing structural problems exactly (analytically) and also by numerical approximation. Now is the time to review the information in Chapters 1–3 and formulate a procedure for using computers as an aid to the structural analyst. First, the key stages of the structural analysis process must be defined, if only in overview; within subsequent chapters these stages will be described in detail. Then the types of computer software that are available will be discussed so that appropriate software can be used to produce the relevant information during the analysis procedure. This will be followed by a discussion of computer hardware architectures and the problem types that can be solved with them. Finally, the ways in which software and hardware are acquired, together with the necessary human skills, will be outlined.

4.1 A PROCESS FOR ANALYSING A STRUCTURE

In Chapter 2, a mathematical analysis of stress and strain led to a series of partial differential equations that govern all structural behaviour. In Chapter 3 functional forms of these partial differential equations were discretized on an element of the structure to produce a numerical analogue of the equations on each element. When boundary conditions and, possibly, initial conditions that define the

structural problem being considered have been applied, the equations can be solved to produce a numerical simulation of the problem.

Regardless of whether locally written or commercial finite element software is being used, the analyst must provide a common set of information to the software before a simulation is possible. From Chapters 2 and 3, this information must include the following:

- A mesh of nodes and elements, developed from the physical geometry being considered, on which the equations can be developed and the variables stored.
- Boundary conditions for loads and displacements (restraints) so that the problem is correctly defined.
- Possibly initial conditions. These might define the initial state of the structure for a transient problem or define the internal forces in the structure due to some initial stress or strain field, or due to body forces such as gravity.
- Material properties such as Young's modulus and Poisson's ratio.
- Control parameters that influence the solution of the equations and the ways in which the results are stored.

For an analyst the process that is followed must first of all generate the above information. Then the process must allow the analyst to check that the results are usable before the relevant engineering information is extracted from the model. One form of the full process can be divided into a series of stages as shown in Fig. 4.1:

- *Initial thinking* Quite often the time taken for an analysis can be minimized if the analyst makes a detailed study of the problem to be solved before touching a computer. This should ensure that the correct problem is solved and useful results are produced. In this initial thinking stage the analyst needs to consider the structural problem and understand as much as possible about it. Here, sources of the geometrical shape of a design, its expected structural behaviour when subjected to loads, typical applied loads and material properties are particularly important. To gather this information liaison is necessary with a variety of people such as design engineers, process engineers, materials specialists and technicians. Further information can also be gained from a literature search of previous work in the area.
- *Mesh generation* This second stage involves the production of a computer model of the geometry of the design being considered, if such a model does not exist already. Then the geometry is discretized to form a mesh of elements and nodes. This means that the structure is broken down into small subregions on which the numerical analogue of the governing equations can be developed.
- *Specification of the numerical problem* In the initial thinking stage, relevant boundary conditions will have been determined in terms of both the physics of the problem and the physical location of the shape. Now, however, nodes and elements have been created in the mesh generation stage and so these boundary conditions must be reinterpreted in terms of the mesh. This will involve the specification of, for example, the forces to be applied at certain nodes, the

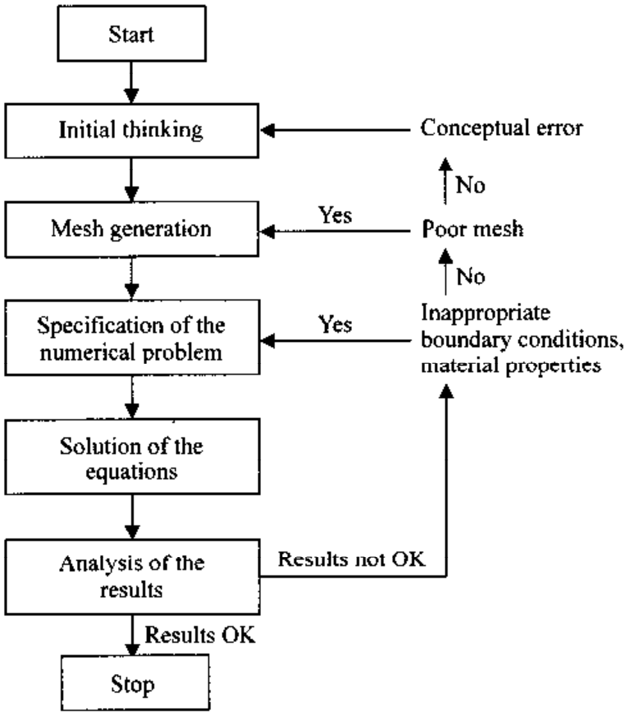


Figure 4.1 The analysis process.

pressures at certain element faces and the known displacements at given nodes. Similarly, initial conditions may also have to be specified in more complex calculations. Finally, the material properties of all the elements will need to be defined.

- *Solution of the equations* Having done all of the above, the solution software can be run. This will calculate a numerical solution for the mesh being considered with the appropriate boundary conditions and material properties.
- *Analysis of the results* Finally, when results have been generated, the analyst must first check to see that the numerical solution is satisfactory before determining the required engineering data from the solution.

While this series of stages might be performed one after the other in a linear sequence, it has already been seen in Chapter 1 that many design activities involve some iteration. This part of the design process is no exception as there are many potential sources of error that might lead to the engineering results being of little use. To reduce the possibilities for errors to occur, a mixture of sensible user experience and good management practice must be demonstrated during the analysis. Here, user experience includes common sense and the development of the art of analysis, with an analyst making suitable choices and approximations

during the analysis. Equally, good management practice includes careful documentation of the work as well as constant checking of progress.

Suitable common checking procedures will be described in the subsequent chapters. If these procedures show that the analysis is not progressing satisfactorily then it may be necessary to repeat some of the stages. Yet again, iteration is necessary if the final solution is to be improved. However, by using computer technology, the refinement of the computational model is usually straightforward.

Common interactions between the stages are shown in Fig. 4.1. Note that internal checking within a given stage has not been shown here, but rather that some typical paths between stages are illustrated. From this, the iterative nature of the process becomes clear, with apparent problems forcing the analyst to loop back to some previous stage in the process in an attempt to improve the solution produced.

4.2 GENERIC SOFTWARE PACKAGES FOR STRESS ANALYSIS

Having looked at the stages of the analysis process, the software that is required to assist the analyst in carrying out the tasks that form each of these stages can now be considered. Taking each of the stages in turn, it can be seen that the first stage requires no computing, the second and third require input from the analyst, the fourth involves computing but no actual input from the analyst and the fifth stage requires the analyst to evaluate the results. From this an appropriate set of software might consist of

- a pre-processor
- a solver
- a post-processor.

Three separate programs could be used, but in most cases the extensive graphical operations required for both pre- and post-processors, coupled with the high levels of user input, lead many systems to have a combined pre- and post-processor system. Also, as much data has sometimes to be transferred from one format to another or from one machine type to another, a variety of utility programs are also used. These relationships are shown in Fig. 4.2.

4.2.1 Pre-processors

All of the tasks that take place before the numerical solution process are called pre-processing. As stated above, the second and third phases of the analysis process involve computing and so the pre-processor software usually assists the analyst in carrying out the following operations:

- definition of geometry in computational form
- definition of a mesh of nodes and elements to represent the geometry

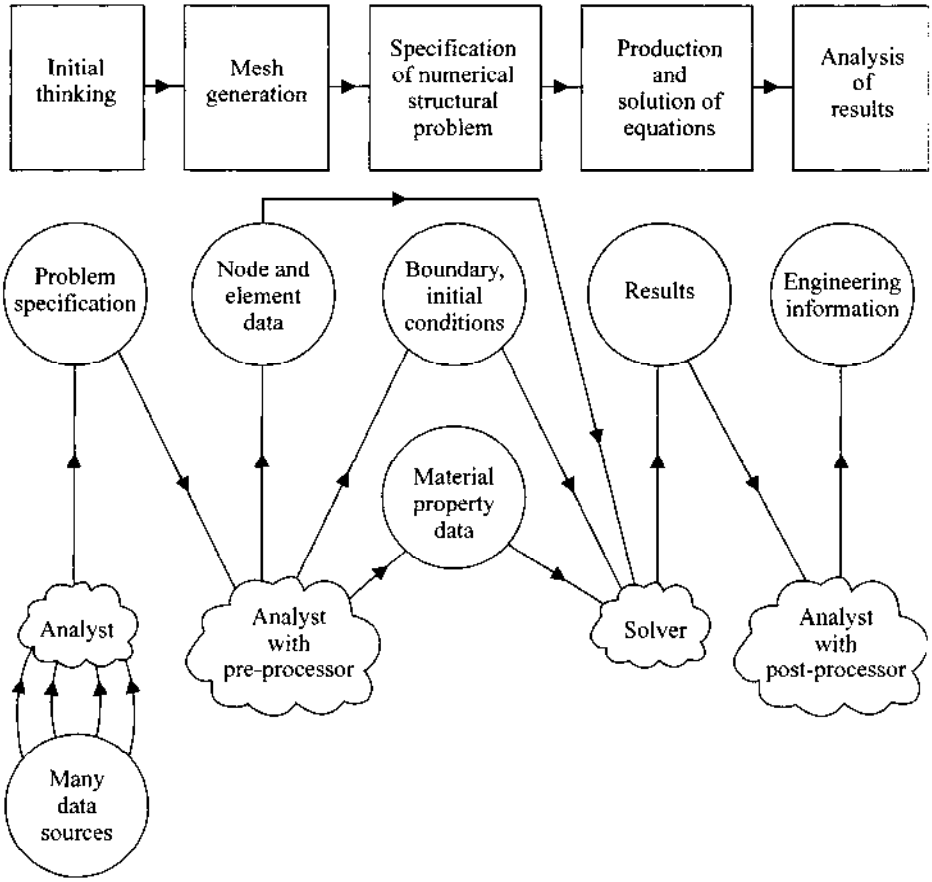


Figure 4.2 Interactions between the analyst and software during the analysis.

- definition of appropriate sections of boundaries of the geometry, in terms of the mesh data, at which boundary conditions will be applied
- application of the boundary conditions
- application, where necessary, of the initial conditions
- definition of material and physical properties for groups of elements
- application of control parameters for the solver.

As all these tasks involve the user interacting with the computer, so pre- processing programs tend to have a user-friendly, graphical interface, probably using the so-called Windows, Icons, Menus, Pull-down Screens (or WIMPS) technology. Using such an interface, coupled with high-quality visual display systems, allows various parameters to be set and the resulting changes to be seen quickly. This is of particular importance when the the geometry of the design is being created and when the mesh is being built. Extensive use of colour is made to assist the analyst.

In many cases, similar designs will be analysed and so files of commands can often be read by a pre-processor and then executed in sequence so that repetitive sets of instructions do not have to be executed by hand each time. Similarly, some systems allow these files of commands to act like computer programs, prompting the user for values and then having suitable logic to act on the input.

4.2.2 Solvers

Although called a solver, this program usually both sets up the required numerical equations that describe the behaviour of a structure under a given set of boundary conditions and solves the equations as well. The solver reads all the relevant data that has been defined by the pre-processor, usually held in files written by the pre-processor, then carries out the necessary numerical operations and writes the results to further files.

If necessary, the files can be moved between computers after being written by the pre-processor and after being written by the solver. This is extremely useful as it allows the solver program to be run on a machine designed to carry out numerical work, 'number crunching', at the fastest possible speeds and the pre- and post-processing programs to be run on machines designed for interactive use. This division of labour between machine types enables the hardware to be used more efficiently.

A further function of the solver can be data-checking. The solver checks to see if the data that it has read is acceptable before attempting to produce a solution. Examples of these checks include ensuring that the coordinates of all the nodes used to define elements exist and that the nodes defining an element are correctly oriented with respect to each other. Clearly such checks might also be carried out when the pre-processor writes the data for the solver.

4.2.3 Post-processors

As has been stated previously, these programs are often combined with pre-processor programs and have a common user interface. As large amounts of information are generated by the solver, graphical interpretation is often the only means of assessing the results. Hence the post-processor is devoted to the display of the results, with typical pictures containing, say, a section of the computational mesh together with vector plots of the stresses, contour plots of scalar variables such as stress intensity, or plots of the displaced shape of a structure when subject to a load. Again, extensive use of colour is made in these programs.

4.2.4 Utility Programs

Often utility programs are provided that are not part of the three programs discussed above. These utility programs usually convert the files of data written by one computer-aided engineering (CAE) system into a format that can be read by another CAE system. For example, information about the nodal coordinates

and the element connectivity is sometimes created by one software package and used by another package that has a different data definition, or results data is written by a solver and then displayed in a post-processor unrelated to the solver. As integrated CAE packages become more widespread so the use of utility programs is reduced. Some people believe they are unnecessary, but at present this is not the case.

Sometimes files that are transferred between systems are referred to as *neutral files*, as they are standard text files that can be written and edited by system editors of many operating systems or by small programs that can be developed locally.

4.3 AVAILABLE COMPUTER HARDWARE SYSTEMS

4.3.1 Computer Architectures

From Secs 4.1 and 4.2 it can be seen that computer hardware is needed to act as a platform on which the software can be run. In the case of the solver program computational speed is important, but for the pre- and post-processing activities large amounts of data must be handled efficiently and computer graphics produced.

This appears to suggest that different computers are needed for the two generic tasks, but in many instances this is not the case as computers are now sufficiently fast that the solution of large engineering problems can be carried out on what might be considered small machines. Even though hardware changes occur regularly, a stable classification of hardware architectures can be used. Essentially, by considering features of the machines, the following five categories can be established:

Personal computers Such computers were developed as standalone systems with their own central processor unit (CPU), some random access memory (RAM) and disk storage. They also have a single-user operating system either resident in read-only memory (ROM) or on disk. More recently these machines have been networked together, with multi-tasking being possible.

Workstations These computers have a CPU, local RAM, true multi-user operating systems and are supplied with a high-resolution graphical display and, often, a reasonable amount of disk storage. Usually they are networked together so that they can access a central data storage system which might be a disk system attached to a *file server* which provides files to the machines on the network. Use of sophisticated networking software and special versions of solvers can enable the machines on the network to act in parallel—a form of distributed processing. Through the network access to high speed computers and peripherals is possible.

Minicomputers These might be considered to be the traditional computer, comprising a CPU, large amounts of RAM and a central disk system. Such machines have multi-user operating systems and are accessed through terminals which have little of their own power.

Minisupercomputers These are workstations with specially accelerated graphics performance and can also provide a good number-crunching capability.

Supercomputers These are designed specifically to be efficient in running large numerical simulations. They consist of very high-speed processors, sometimes in multiple units, connected to a vast RAM storage system with fast access to reduce the time spent communicating with slow-speed disk storage.

Novel architectures This category includes several types of machines, but two are predominant. Both use an array of processors, with one having a small number of expensive and fast processors and the other having a large number of inexpensive but slower processors. They are known as parallel machines, with the latter type having the special designation of massively parallel machine. Here a trade-off is made between computing power and cost, with the same total amount of processing power provided on both types for much the same overall cost. Unfortunately, such machines are at the forefront of technological development and so very little commercial stress analysis software will run on them.

Figure 4.3 shows the main architectures that have been described, but as technology changes, so the boundaries become increasingly blurred.

4.3.2 Peripherals

To achieve a good operational turnaround during an analysis, the user must rely on the total system, not just the raw computing power available. Peripheral devices such as the following can help speed up or slow down the process.

Secondary data storage devices All engineering analysis programs that involve the numerical solution of equations such as the stiffness equation (1.2) throughout a multi-dimensional domain are required to access and store large amounts of data. Where possible this data is held in RAM storage during program execution as the response time is then the fastest available. However, secondary storage devices such as hard disks are also used both during execution when the available RAM is not large enough, to prevent repetitive calculations, and to enable data to be available to post-processors. As access to secondary storage is slower than with RAM its use can reduce the efficiency of any of the program types quite dramatically.

Backup devices These are used to protect the data from being lost. This can occur when hardware, such as a disk drive, fails or when accidents, such as a machine-room fire, happen. Users, or more likely, system administrators, must

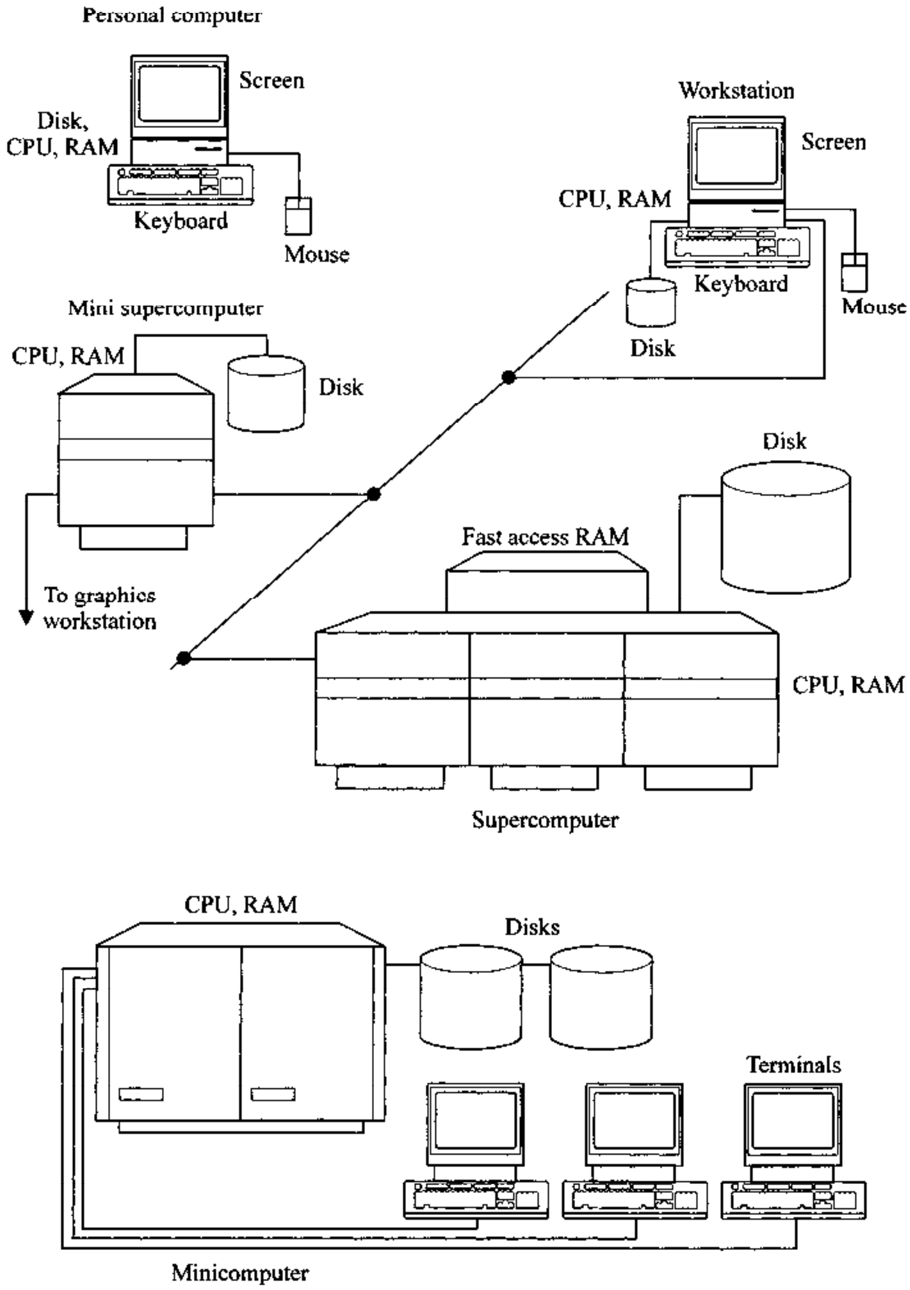


Figure 4.3 Types of computer hardware.

make regular copies of the data onto some form of backup data storage device. Typically, demountable hard disks or magnetic tapes are used as these can be removed to a safe storage area somewhere remote from normal operations.

High-resolution graphics displays As has been stated, graphical analysis using computer-generated displays in colour is vital to the success of large stress analyses. For user comfort high-resolution graphics display devices are used as the interactive display medium. Typically, the resolution of these devices is of the order of 1000 *pixels* in both directions, where a pixel is one dot of an array on the screen.

Hardcopy devices While looking at a screen is important to the analyst, hardcopy is also required so that sensible communication with others can be carried out. Such hardcopy is usually created using either laser printers which produce monochrome or colour paper copies or plotters which use ink jets or heated waxes to produce a coloured image.

4.4 MATCHING HARDWARE TYPES TO ANALYSES

Now an attempt can be made to match hardware and software for a variety of types of finite element analysis. First of all, computers can be classified by some measure of their calculation speed. This can be measured in units based on the number of internal processor instructions executed per second or the number of floating-point operations achieved per second. Common units are *mips* (millions of instructions per second) and *MFLOPS* (millions of floating-point operations per second). While these measures give some idea of the speed of a machine, they do not consider the complex interactions between a computer and the program being used. Such interaction is very important as the speed of reading from or writing data to RAM or disk can be just as important as the speed at which the numerical calculations are carried out. Further, each individual program is written such that the number of calculations is different for the same overall calculation and the amount of data access also varies. Hence it is very difficult to draw up hard and fast rules as to the turnaround time for an analysis.

Other considerations include the ways in which graphics data can be handled, in terms of both the speed of producing images and their resolution, as well as the amounts of data that can be stored. By considering these requirements—data storage, processing power and graphics capabilities—each of the computer architectures can be considered in turn to see to which types of analysis they might be best suited. For example:

- Personal computers (PCs) often lack fast processor speed or data storage capacity. Hence, they tend to be used for smaller problems that are simple to compute, such as linear statics analysis. An ideal situation might be in training analysts, where small demonstration problems are run and so the turnaround time is negligible.

- Workstations (or high-specification PCs) have the computing power, data storage and graphics capabilities for most finite element stress analysis problems where linear statics analysis alone is carried out. Equally, when optimization of a structure is required or dynamic or nonlinear behaviour is to be modelled, many problems can also be solved. It is only when very large and complex analyses are required that faster machines tend to be used.
- Minicomputers perform in a way similar to workstations although their graphics capability is often poorer.
- Minisupercomputers are versatile in that they can be used for most problems, even large and complex simulations. This is especially true if the turnaround time is not too important.
- Supercomputers are mainly used for the largest and most complex problems.

Table 4.1 shows a summary of these points. Here the graphics capability of the hardware types is rated together with the applicability of using these types of hardware for small, medium and large analysis jobs. Clearly, the definitions of the size of a job are somewhat arbitrary, but small jobs could be those with only a few hundred to a few thousand nodes for linear statics analysis of say a two-dimensional or axisymmetric situation. Medium jobs could be those with a few tens of thousands of nodes for linear statics analysis and large jobs might be those with tens of thousands of nodes or more and those where more complex analyses such as time dependent or nonlinear simulations are being carried out. Note that some structural analysis can, in most cases, be carried out with only a limited amount of hardware.

4.5 ACQUIRING THE TECHNOLOGIES

4.5.1 Some Preliminaries

By using the hardware and software that is commercially available, designers have access to a tool that is complementary to physical experiments, semi-empirical methods and analytical techniques in analysing proposed or existing structures and might lead to cost reductions. However, the acquisition of the necessary hardware and software, as well as the appropriate human expertise, is expensive.

Table 4.1 Suitability of hardware types

Hardware type	Graphics	Small jobs	Medium jobs	Large jobs
PC	OK	OK	No	No
Workstation (or high-spec. PC)	Good	Good	Good	No
Minicomputer	OK	Good	Good	No
Minisupercomputer	Good	Good	Good	OK
Supercomputer	No	OK (expensive!)	Good	Good

For this reason, a careful study of the needs and requirements has to be made before the final decision to acquire the technology can be taken.

4.5.2 Choosing Software

The first thing to consider is how a knowledge of structural mechanics might help you or your organization. To explore this, all the functional areas that are related to structural mechanics must be considered. Take the case of a manufacturer in the motor industry. Structural analysis may be used to determine the linear static stress and displacements in structures such as the vehicle body shell and the engine under operational loads. Also, optimization may be required to produce body shells with a given displacement for the minimum material thicknesses. Further, coupled heat transfer and structural analysis may be used in analysing the distortion of the braking systems and the engine block and, finally, crash simulation of the body shell may be required.

Once the areas of interest have been listed, the techniques that are available to investigate the particular type of analysis should be assessed. If computational tools have a place in the toolkit of the designer then the benefits of using the technology must be made clear. Remember that computational tools may not provide more information than experimental procedures, but they may provide some extra benefit, such as a direct saving in financial terms through the reduction in the number of physical models required, or the ability to carry out more tests in a given time.

If there is a clear need for structural analysis software, then the type of software that is required must be decided upon. To find out which of the available packages may be used, a list of requirements that the software should meet must be produced. More often than not, no single package will meet all the requirements, but several packages will meet some of the requirements. Hence, when choosing the software analysts may have to make some very subjective decisions.

To draw up the program specification, analysts must think carefully about the structural problems that they wish to analyse and the priority that is to be attached to them. For example, consideration might be given to the following features:

- *The geometry of the structures that may need to be analysed* This will show whether a package is needed that can solve problems in two or three dimensions. Invariably, three-dimensional solutions will be required, but the analyst needs to see if the geometry is very complex, when an unstructured mesh is required.
- *The structural analysis type* Here the analyst must decide which type of analysis needs to be carried out. This could be simple linear static analysis or it might include a requirement for optimization. Equally, the simulation may need to be time dependent or have to calculate the effects of nonlinearities such as plasticity and/or buckling.
- *Coupling requirements to other software* In some cases there may be a need to link structural results to heat transfer simulations or even to fluid flow soft-

ware. There are also other interfacing requirements. For example, the software may need to have access to geometrical data created or stored by a proprietary CAD system. Equally, there may be a requirement to send the results to a proprietary post-processor or to some other display software. Clearly, in these cases, the structural software must have appropriate interfaces.

- *The size of the simulation problem* Here something about the number of nodes and elements that a typical mesh contains needs to be known, together with the number of degrees of freedom that are to be calculated. This information helps to determine the storage requirements of the programs in terms of both primary and secondary storage, as discussed in Sec. 4.5.3. Remember that the more data that is calculated, the more accurate the solution should be, but the longer it will take to obtain the results. Clearly some compromise has to be made here.
- *The required results of the analysis* Stresses, strains and displacements, possibly as a function of time.
- *Solution speed* Many things affect the time that it takes to produce the solution. Clearly, this depends on the processing speed of the hardware that is used, but it also depends on the structural solver itself. Some algorithms for solving the governing equations are much faster than others. This speed difference can come from the basic discretization of the equations, from the internal organization of the program or from the speed of the linear equation solvers used.
- *Hardware availability* If there is a restriction on the make or type of computer or graphics terminal that the software can be run on, this should be noted. It is common, however, for pre- and post-processors to be very hardware specific and for solvers to be much more portable between different machine types.

From all of the above, a considerable amount will be known about the situations that are to be simulated and the simulation process itself. Finally, it is important to assess what kind of service it is that the software supplier must provide. This is a very subjective set of requirements, and to some extent depends on the people who are available within an organization to run the analysis software and talk with the supplier. At this point, it is worth issuing a warning. With the proliferation of computers and software, many people are now used to buying a package, loading it onto a machine and getting results without too many problems. For business software this is certainly true, and it is becoming true of many engineering packages as well. Unfortunately, any software that solves partial differential equations will not, by its very nature, be as mature as other simpler engineering analysis tools that are on the market. Hence, in most cases, the learning curve will be very steep and costly.

The following are some of the requirements that are related to the software supplier:

- *Quality assurance (QA)* This is the extent to which the software has been tested against standard test cases for which there is a known solution. The comparison data may come from either analytical expressions or experiments.

These comparisons are carried out to both verify the code, i.e. to show that it is correct against the numerical models that it is simulating, and also to validate the code, i.e. to show that the code gives reliable results against physical experiments. There should be some evidence from the supplier that the software has been tested for both validation and verification. In fact, most pieces of structural analysis software are so complex that every possible combination of operating features will never be tested until, that is, you as a user run your particular example and find that it does not work. This may appear cynical but it is often true.

- *User friendliness* This is probably the most subjective feature of all, as what appears friendly to one person may be unfriendly to another. Again, this depends on the staff who run the programs. However, it is worth looking at the user interface to see if the menu structure is logical.
- *User support* This is very important as users can never be fully conversant with the programs that they run. Software suppliers should provide some form of user hotline that can give a quick response to a user's questions. This normally comes as part of the annual licence fee for the software, or can be purchased separately if the program is bought with a once-only payment, which is known as a perpetual licence. There should also be the option of buying training in the use of the programs, and the chance for users to work with the supplier in setting up a problem. This is normally done by paying for consultancy from the software supplier.
- *Current users* It is important to know who *is currently using* the software, not who *has used* the software. This enables companies to see if firms in a similar business are using the software, and can give some confidence in the software supplier and their product.

The easiest way to document this information is to draw up a table of capabilities product by product. Sometimes this can be difficult to do for someone with little experience. It is at this stage that it is important to obtain independent advice to guide you. Sometimes people place too much reliance in the software suppliers themselves and, even though the suppliers can provide much information, an independent view is worth while. A sample specification table is shown in Table 4.2 and this can be used to assess each of the competing products.

Once the specification of the software is determined, various software options need to be evaluated against this specification. This can take a lot of time and effort as there are many products in the market-place and the suppliers of each of them will be only too willing to shower you with information. The information that is provided can take many forms, but the simplest starting point is to look at the brochures that explain the software. Much of the information required can be determined from these, but quite a lot of it cannot. In particular, the more subjective information such as the levels of user friendliness, solution times, QA and user support need to be investigated further.

One way of gaining this more specific information is to produce a sample problem that is typical of those you wish to solve. Suppliers will often produce a

Table 4.2 Assessing a program specification

Capability	Prog. 1	Prog. 2	Prog. 3	...	Prog. N
Dimensions (2 or 3)					
Mesh type (structured or unstructured)					
Linear statics (yes or no)					
Normal mode dynamics (yes or no)					
Buckling (bifurcation) (yes or no)					
Forced dynamics (yes or no)					
Large deformation (yes or no)					
Plasticity (yes or no)					
Optimization (yes or no)					
Time dependent (yes or no)					
Heat transfer (yes or no)					
Coupling to fluid flow (yes or no)					
Limits on model size					
Interfacing to CAD					
Speed of solution					
Hardware availability					
Evidence of QA					
User friendliness					
User support					
List of users with similar structural problems					

simulation of this problem using their software at a reduced cost, or even for free if the problem is very small. This enables potential customers to see software products in action on a realistic problem. Such a trial helps in understanding how the processes outlined in this book relate to the specification and operation of the software. It also produces some hard facts that should help in determining the cost of obtaining a simulation using a particular package.

When the competing products have been assessed using the specification table, several suitable products should emerge. It is probable that none of the products will be ideal, but some should come closer than others. A simple way of assessing the most suitable package is to assign numbers to each of the categories in the specification in some way such that the higher the number the better the specification level. By adding up these numbers and getting a total value for each package they can be ranked.

Once the products are ranked in order of suitability, the question of cost needs to be looked at. Normally, software is licensed on an annual basis with a single fee being paid to the supplier which includes the provision of the software and any updates to it as well as technical support in the form of a hotline service. Sometimes, however, the software is purchased on perpetual licence terms where

one large payment pays for the software and a smaller annual fee pays for the updates to the software and the support. Sometimes both methods are on offer, and it takes careful consideration to decide which of the two will be the cheapest option in the long run. This is especially difficult as the market is still developing and the most suitable program today may not be the best choice in three or four years. Finally, it may be that some sacrifice in terms of the capability of a package has to be made if an affordable solution is to be chosen for purchase. This is achieved by determining the minimum level of functionality that is acceptable.

4.5.3 Choosing Hardware

Many organizations already have access to the computer facilities that are necessary for running large computational analysis programs. Others will need to acquire the hardware. In both cases, however, it is important to consider the following factors. For the former case these factors enable the user to determine if the existing facilities are suitable and have the necessary spare capacity, and for the latter case they allow estimates to be made of the various measures that will determine the hardware.

Computer processing power A large amount of processing power is needed to run some structural problems. Fortunately, recent technical advances mean that the necessary power is available cheaply. The factors that affect the speed of processing include such things as the calculation speed of the processor which is measured in mips or MFLOPS. There is no clear relationship between the two for different processors, as what takes one instruction on one processor might take several instructions on another. The speeds of the various computers are often quoted in these units, but different software runs in different ways on different machines. Consequently, the numbers quoted are only a guide to the raw processing power. To find a true measure of speed for the software and hardware combination a series of sample problems must be simulated. This assumes that the software does not make any use of the secondary data storage during execution, as the speed at which data can be accessed from devices such as hard disks can have a marked effect on solution times. Some software packages write data to these devices during the solution phase and if the processes of reading from and writing to the disk are slow, then the whole solution process is slowed down. For a typical analysis on a given computer installation, the total solution time depends on all of these things together with the number of simulations that are solved simultaneously on any one system.

Primary data storage capacity On most computer systems the primary data storage system is RAM. This is usually sized by the number of bytes of data that can be stored. Each byte consists of eight bits, where one bit is the basic unit of storage corresponding to a stored value of either zero or one. Numbers can be stored as integers or real numbers and two or four bytes are used for integers

and four or eight bytes for real numbers. The greater the number of bytes the greater the maximum integer, and the more accurate a real number, that can be stored. Sometimes, the software supplier specifies the number of megabytes of RAM that are required to run their software successfully. In large machines, such as supercomputers, the memory size is measured in words. These are usually words of eight bytes or sixty-four bits, and are the machine's minimum storage for a single real number.

Secondary storage capacity This is usually provided by hard disks, which are aluminium disks covered in magnetic material such as iron oxide, just as with audio tape. In personal computers these disks may store tens of megabytes and in workstations several hundred megabytes. In large systems, the disk storage might consist of sets of disks each storing several gigabytes of data. Analysts need to assess how much of this storage is needed for each problem to be solved. A rough estimate can be made by taking the number of nodes in a problem and multiplying this by the number of coordinates used to describe the spatial position of a node plus the number of variables stored at each node or element. For a three-dimensional linear statics problem there are 3 coordinate directions and 3 displacements per node giving 6 variables at each node. So for a mesh with 10 000 nodes, at least 60 000 real numbers must be stored. If the data is stored in readable (ASCII) format, say 20 bytes (or characters) are required to store each number, and therefore 1.2 megabytes are required in total. If, however, the data is stored as single precision real numbers in binary format, only 4 bytes are required to store each number and the total storage required is 0.24 megabytes. These are low estimates of the total data storage requirements, as each software package will store different information, but a software supplier may be able to give information on the data storage required for a given model size.

Access points If several people need to run analyses simultaneously then several access points are required. These may need to be split between a number of graphics screens and a number of text screens. This enables some people to perform graphics pre- and post-processing, while others run the solver program.

Backup facilities There is a need to provide some backup of the data held on disk, to protect against loss of data. This can occur if a disk drive is broken, such that the data stored on it cannot be read, or can occur if a user deletes a file in error. It is common for each disk to be backed-up in full, i.e. all the data is written to a tape storage device, or something similar, every week. Further backup procedures are carried out once a day, to ensure that all the new files that are created within the previous 24 hours, and the new versions of edited files, are also written to a backup device. This procedure is known as an incremental backup and ensures that, at worst, only one day's work can be lost. Once backup tapes have been prepared it is worth protecting them against fire by using a fireproof storage facility.

When these items have been considered, it should be possible to know whether an existing installation will be sufficient to run the relevant structural problems or whether it will need to be enhanced in some way. If new facilities are required, either to enhance the existing capacity or to provide a completely new system, then they can now be assessed for suitability.

4.5.4 Building an Analysis Team

Having decided upon a software package and a hardware system, it must be made clear that the simulations do not run themselves. Consequently, some consideration must finally be given to the most important asset in the analysis process. This is the analyst who actually translates the engineering problem into a computational simulation, runs the solver and both checks and analyses the results. It is the skill of this person, or set of people, that will determine whether all the hardware and software is utilized in the best possible way and so produce good quality results.

To produce a successful and cost-effective linear static finite element model the analyst has to be capable of the following:

1. Understanding the fundamental physics of the problem so as to gain a good impression of the expected solution. Here physical interactions between the structure and its environment are the principal difficulty. A working knowledge of classical elasticity, as presented in Chapter 2, is most useful as it allows the analyst to determine where in the structure the stresses, and their gradients, will be highest. Load paths through the structure can also be visualized. Also the analyst needs the ability to identify regions of stress concentration and determine die-away lengths (using Saint-Venant's principle) associated with a given type of discontinuity.
2. Having the skill to choose the relevant dimensionality and the structural form for the preparation of a satisfactory model. This also requires an understanding of the limitations of the applicable theory and so, again, there is justification for knowing the theory of classical elasticity.
3. Understanding the relevant facilities of finite element packages and how these are used to model the various features of the physical problem.
4. Understanding the behaviour, accuracy and limitations, such as 'locking' and mesh instabilities, of the various finite elements available, the approximations made in representing real boundary conditions (loading and restraints) and the solution algorithms.
5. Having the skill to construct a mesh with the minimum number of degrees of freedom that provide the desired accuracy while preventing round-off errors from becoming intolerable. This includes the application of techniques such as assuming symmetry, assigning multi-point constraints and substructuring.
6. Having the skill to choose relevant and reliable material property data and, if required, failure criteria for the material used in the structural problem.
7. Having the skill to develop a reliable and relevant model that meets the goals of the analysis.

8. Modelling the real boundary conditions of loadings (mechanical loading, thermal loading, displacement conditions or initial conditions), supports, constraints, joints, gaps and releases, and ensuring that all rigid body motion is removed without altering the structural response.
9. Checking, before the solver is run, that the model is valid.
10. Interpreting output results sensibly and without reliance on powerful graphic post-processors that can smooth out stress discontinuities that may be warning the analyst that the model is wrong.
11. Comparing solutions with whatever else is available, such as 'back-of-envelope' calculations, approximate analytical models, experimental data, textbook and other computationally generated results, which may well be based on a different numerical method.
12. Refining the model using h- , p- or adaptive mesh refinement with the minimum of work to determine the rate of convergence to the 'exact' solution.
13. Using the results from the analyses to make the 'correct' design decisions, what ever they may be.

This comprehensive list of skills and knowledge that an analyst requires to be a successful user of static finite element analysis in the design process illustrates the many difficulties that have to be faced. Many of these will be addressed in later chapters. Note that if the structural problem requires analysis of dynamics, buckling and/or nonlinearity, then the list of skills and knowledge will be increased significantly from that given above.

People from many different backgrounds can be trained to carry out the processes listed, but it is our belief that more than this is required. The successful application of the finite element method (FEM) requires not just basic knowledge but a considerable degree of experience on the part of the analyst which can only be obtained by practical use of the finite element model.

Another way of looking at the skills required is to consider broad categories of skills across four areas.

Mathematical skills These enable the analyst to understand the underlying features of the numerical processes used to convert the governing partial differential equations into numerical analogues, and to coax the solution procedures to converge to sensible and realistic values.

Computational skills The production of a simulation can involve the user in manipulating large amounts of data with packages that do not interface together and reside on a variety of types of computers. This can mean, for example, that analysts have to write their own interface programs to convert data from one program's format to another program's format. Also, an analyst might have to write computer operating system command language programs that instruct a computer or even a variety of computers to move data around a network, run some programs and then move the data around the network again. Consequently,

analysts must be conversant with computer procedures at a level that is far greater than that required for analysts who use the more common software products that perform engineering computations.

Good interpersonal skills If the analyst is not the end user of the data, then there has to be close liaison between the analyst and the end user, who is in effect a customer or client of the analyst. This requires that a good working relationship is developed between the two parties so that the analyst knows what the customer requires, and the customer is aware of the limitations of the analysis.

Engineering skills Finally, the analyst must have a working understanding of the engineering processes that are to be modelled. This enables the limits of a computer model to be established and the results of the simulation to be analysed in a sensible way.

Large organizations may well have a pool of analysts in which there are several people who could be used to produce structural simulations, as they have a majority of the qualities listed.

If such people do not exist within the organization, or if suitable people cannot be used for whatever reason, then staff will have to be hired. Hiring staff of the right technical background to use finite elements in industry, whatever their background, is extremely difficult. Not many people have all the skills necessary and so several people may be needed. Depending on the size of the organization, therefore, one or more people may be employed in the use of finite elements, and the right mixture of abilities is important.

One other way of proceeding is to employ a limited number of people to work with finite elements and then to use external consultants to supplement the skills where appropriate. These consultants can be found working with software suppliers, general engineering consultancy practices and in universities. For industrial users who are not specialists in this field, it is important to have access to advice at a moment's notice. This can be provided by a software supplier when problems occur running a particular package, but a local university may well have a specialist in the finite element field who may be willing to provide consultancy as and when required.

5

BUILDING THE GEOMETRY DESCRIPTION AND THE MESH

This chapter contains material previously published in Shaw (1992), by kind permission of Prentice Hall.

5.1 THINK FIRST—COMPUTE LATER

When using a computer to simulate any engineering problem an analyst must specify the model in the form of data to the solver. This data must reflect the actual engineering problem being solved. While the data must be both suitable and accurate, it may be that other factors such as a user's experience will determine the quality of the simulation. Given that this is the case, an analyst must become familiar with the engineering problem that is to be simulated. While it is a common failing for those who use computers to be tempted to start computing as soon as possible, it is much better if careful thought and planning is given to the problem first. Hence the urge to compute before thinking must be resisted.

Particular problems arise when people use computers as black boxes. The well-known computer acronym GIGO applies here—garbage in, garbage out—so users of finite element packages for mechanical design should be thoroughly familiar with the material in Chapter 2 before they embark on a static analysis. In particular, the thinking required to develop a good feel for a problem is given in Sec. 2.4, where a problem specification is produced that is the main source of information for the analyst. It is the information contained in the specification that has to be translated into terms that the software package can understand.

Once the specification of the structural problem is known, attention can be turned to the next stage of the analysis process, where the mesh is generated. When modelling a simple problem this process takes very little time but, when modelling a complex geometrical problem such as the body shell of a motor vehicle or a casting for a piston engine, the process can take one engineer several months to complete. Often it is this stage of the analysis process that determines the total time required to obtain results from a simulation, as all the other phases, including the actual computation of the results, can be carried out quite quickly. Similarly, the overall cost of the analysis can be totally dominated by the costs of the labour required to build the mesh.

In fact, the mesh generation stage can be subdivided into the building of a computer model of the geometry that is suitable for use in the mesh generation process and the creation of a mesh to represent this geometry. Both subdivisions will be covered in this chapter. First, the ways in which geometry can be modelled using computers will be discussed before looking at the engineering approximations that can be made. Then the reasons for building a mesh, the choice of appropriate element types, the requirements that a mesh must satisfy if it is to give satisfactory solutions and the types of mesh that can be built will be discussed. Finally, the ways in which a mesh can be built using a variety of software tools and the ways in which a mesh can be modified in the light of the results of a simulation such that better results are achieved will be considered.

Throughout this chapter it should be kept in mind that it requires considerable skill and a good deal of experience to be able to construct a mesh that will produce an accurate solution for an acceptable cost for even a moderately complicated structure.

5.2 OVERVIEW OF COMPUTER MODELS OF GEOMETRY

In the specification stage of the process that was discussed in Sec. 2.4 it was seen that the sources of geometrical data must be determined and sketches produced of the structure. For structures with essentially one-dimensional elements such as bars and beams this includes the positions of these elements, but for more complex structures this includes the positions of the bounding surfaces. These sketches must be used together with the sources of geometrical information to produce a computer model of the geometry.

While it is not necessary to understand the mathematics behind the descriptions, the analyst should have some knowledge of the variety of computer models that can be used. In particular the following three levels of model are often used:

- wireframe models
- surface models
- solid models.

The level of complexity increases when going down the list, with the first handling just curves, the next handling surfaces and the final one handling the

internal volumes as well. For those who are interested, several books describe the ways in which these numerical descriptions of objects are handled in CAD systems, for example, Rooney and Steadman (1987) and McMahon and Browne (1993).

5.3 WIREFRAME MODELS

A special case that needs to be dealt with here is the class of structures that can be thought of as consisting of one-dimensional elements. These are special as the geometry is completely described by the end coordinates of the bars and beams, together with the connections between them. Hence while computer models are of the wireframe type, they are different in nature to the classical wireframe models.

For two-dimensional problems, the bounding surfaces can be created using points to define a series of lines and curves. These curves might be defined as circular arcs, simple polynomials or splines. All of these constructions are described by equations that define the relationship between the coordinates of points that make up the curve. For example, it is known that a line can be described by the relationship

$$y = mx + c \tag{5.1}$$

where m is the gradient of the line and c is the value of y when x is zero. By substituting for the gradient and intercept in terms of two known points on the line, (5.1) becomes

$$y = \frac{y_2 - y_1}{x_2 - x_1}x + \frac{y_1 x_2 - x_1 y_2}{x_2 - x_1} \tag{5.2}$$

where the subscripts refer to the two known points. These equations describe a line that is infinite in length but, as lines of finite length only are used to describe the geometry of a plane structure, the endpoints of the line must be known.

Similarly, a circle can be described by

$$(x - a)^2 + (y - b)^2 = r^2 \tag{5.3}$$

where the centre of the circle is at $x = a$, $y = b$ and the radius is r. A part of this circle is a circular arc and three points can be used to define it. Usually these points are taken to be the two end points of the arc and a point on the arc somewhere between them. This enables both the limits of the arc to be defined and the unknown constants in (5.3), namely a, b and r, to be calculated.

Splines are more complex curves but they are also defined by points in space. Usually four or more points are used, and these do not have to be on the curve itself. Note that there is a hierarchy being formed here in terms of the numbers of points required to define a curve. Two points define a line, three points define an arc and four or more points define a spline.

Clearly, in three-dimensional problems, a similar hierarchy exists and so points in x, y and z can be used to define three-dimensional lines, arcs and curves.

These in turn can be used in the form of a *wireframe model*. In such models key edges of the model are defined by points in space, i.e. the model consists of a series of edge–vertex relationships. Here, several lines, arcs and curves can make up a single edge.

Typical uses of this type of geometry model in mesh generation are for simple meshes in two dimensions or for simple three-dimensional meshes such as a sheet of nodes and elements through space, as might be used to model a topologically rectangular region of the surface of a structure.

5.4 SURFACE MODELS

In many cases a wireframe model is not sufficient for mesh generation as detailed information is required as to the bounding surfaces of the geometry. While it is possible for a structure to have surfaces that are of simple form, such as a plane or part of a sphere or cylinder, the surfaces of a structure often have a more complex form. When this is the case the complex surface is often described by a series of simpler surfaces which are local approximations to the physical surface.

Typical approximations to complex surfaces are simple patches, Coons patches, Bézier surfaces and non-uniform rational B-spline surfaces.

Numerous simple patches Here the surface is discretized into a series of patches that are usually triangular or quadrilateral in form. This is the way that a surface is described by the faces of a mesh of linear elements.

Coons patches These are patches over which the coordinates of points on the surface are determined from the bounding curves alone. Consequently, once the boundaries of a surface are determined the surface itself is defined. Three or four curves in space, or assemblies of curves (i.e. edges), which form a closed loop are often used to define the boundaries. Note that an infinite number of surfaces can fit through a given set of boundaries but that the Coons patch description defines only one such surface. The assumption has to be made that each patch represents a sufficiently small part of the surface so that a good approximation to the surface is given. This approach can lead to problems if a surface is highly curved and only a few Coons patches are used to model it. In this situation each patch will be too large and the surface definition will depart dramatically from that required.

Bézier surfaces These are surfaces described by a set of Bézier polynomial curves. Each curve is defined by four points, the two end points of the curve plus two interior points which need not be on the curve. By moving the two interior points the curve can be manipulated to have a wide range of shapes. To construct a Bézier surface, 16 points are used to define a 4 by 4 lattice of Bézier curves and by interpolation between these curves the surface coordinates can be found. Bézier surfaces give an improved description of a surface when compared to a Coons patch description, as information from within the boundaries is used to define the

surface. This helps to lock a surface in space and so the number of surfaces that can fit the description is reduced. These surfaces were developed for Renault, the French vehicle manufacturer, as they required computational surfaces that could be manipulated interactively when modelling new vehicles in the styling studio.

Non-uniform rational B-spline surfaces (NURBS) These are similar to Bézier surfaces, but the curves that are used to define them are based on different points to the Bézier curves. The end points of the curves are only approximated but the points that are used to define the polynomials ensure that the spatial derivatives of the first- and second-order are continuous at the end points.

5.5 SOLID MODELS

Many CAD systems now incorporate both wireframe and surface models together with the third model type, the *solid model*. Here, not only geometric information but also such data as the mass of an object is stored. To produce this information two main types of methodology are used, *constructive solid geometry (CSG)* and *boundary representation (B-rep)*.

5.5.1 CSG

In CSG complex objects are built from combinations of primitive objects such as cylinders, tubes, blocks and cones. By performing the Boolean operations of union, intersection and difference, a graph or tree data structure can be developed. For example, Fig. 5.1(a) shows a thick plate with a cylindrical hole through it. This object can be considered to be built up from the union of two plates and the difference between the union of these plates and a cylinder as shown in Fig. 5.1(b). This data structure is known as a CSG tree.

5.5.2 B-rep

In B-rep models the objects are defined by their boundary descriptions. For example, a closed loop of curves, also known as a profile, might be extruded through space to form a three-dimensional object, as shown in Fig. 5.2(a), where an extrusion along the z-axis is carried out for a profile defined in the x–y plane. Another common operation is to sweep a profile through some angle to form a volume of revolution, as shown in Fig. 5.2(b).

More complex B-rep models can be created by taking a series of profiles that have been defined along some path in space and then effectively placing a surface over these profiles. This operation is known as skinning and an example is shown in Fig. 5.2(c) where three circular profiles are placed with their centres along some path in space and a skin thrown between them. The process works in a similar

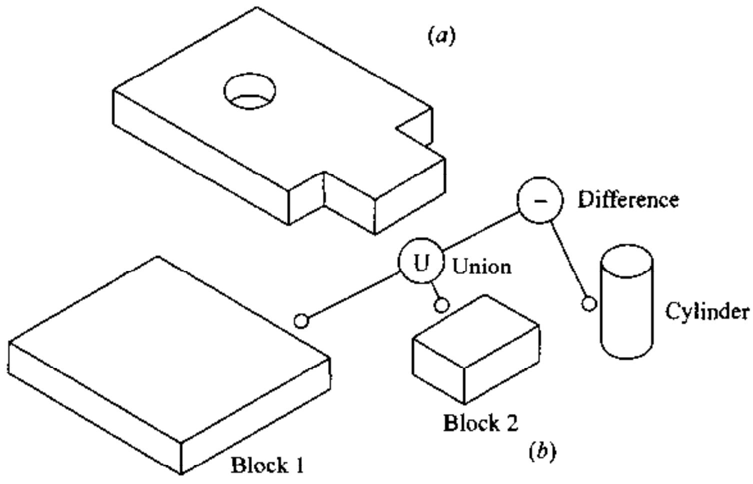


Figure 5.1 A CSG model of a plate with a hole: (*a*) the final object, (*b*) the CSG tree of Boolean operations.

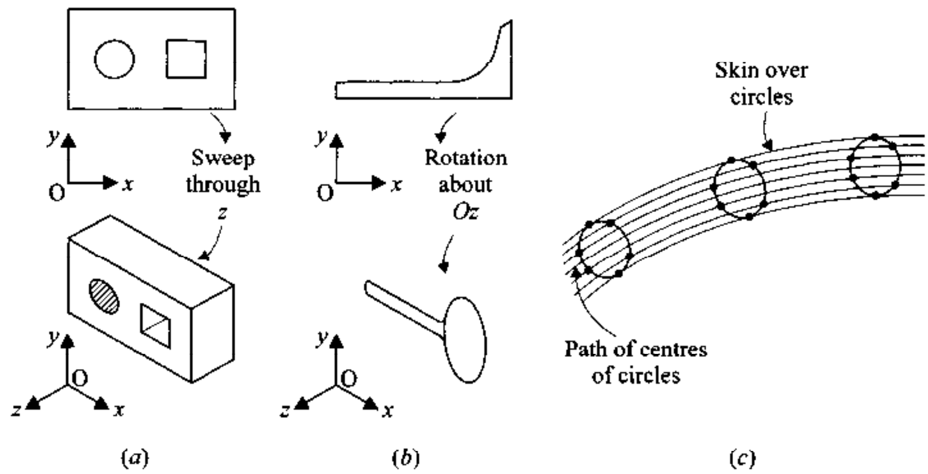


Figure 5.2 Operations for B-rep models: (*a*) sweeping a two-dimensional profile in the third dimension, (*b*) forming a solid of revolution, (*c*) skinning across a set of profiles.

way to the lofting of ship hulls where complex curves are placed along the length of the ship to define the hull shape.

5.5.3 Commercial Solid Modellers

Most commercial solid modellers combine the features of CSG and B-rep to give a more flexible modeller. Such systems allow CSG-type operations to be carried

out on a range of solids which can be both primitives and B-rep models. Occasionally, other types of solid model can also be used such as pure primitive instancing and cell decomposition (Rooney and Steadman, 1987).

5.6 CHOICE OF GEOMETRY DESCRIPTION

Several factors affect the way in which the analyst chooses to describe the geometry. These include the following:

- The *original sources* of the geometry data. For example, if there is no computer description of the geometry the analyst is free to use whichever description is appropriate, subject to the factors that follow. Equally, if the geometry data is part of a computer database already, the analyst will want to use that data in as efficient a way as possible, obtaining the appropriate information with the minimum amount of effort.
- The *tools available* to the analyst. Clearly, these might restrict the approaches that the analyst can take if software is not available.
- The *physical problem* being modelled. Where the use of dimension reduction to model a three-dimensional problem as a one- or two-dimensional problem or the use of symmetry can be very effective in saving computer effort.

It is the last of these three factors that can make or break an analysis. Remember that three-dimensional analyses are at least an order of magnitude more expensive than two-dimensional ones and that, because there are many more equations to solve, such simulations are prone to round-off errors. Further, it is found that three-dimensional models present problems in mesh generation, because of large element distortion, and in checking, because of poor element connection providing unwanted cracks that are difficult to detect and rectify. Additionally, while the subdivision of a two-dimensional domain by triangles is easy to visualize, the subdivision of a volume by tetrahedra is extremely difficult to visualize.

As a guide:

> Use three-dimensional models only if either one- or two-dimensional approximations have been found to be unacceptable.

Equally, the use of symmetry is advantageous in that it leads to a reduction in the effort required for model development, a reduction in the required computer power, computing time and cost and to a decrease in the computer round-off error when solving the equations since fewer equations exist in the model. However, there are disadvantages in the use of symmetry: it can be more difficult to picture the model and the peak stresses may occur along symmetry surfaces making it more difficult to locate the peaks accurately.

If the problem has symmetry of material properties and boundary conditions then one or a combination of the following four types of geometric symmetry can be exploited:

- *mirror symmetry,* where the geometry can be modelled by one half of the full geometry as there is a plane of symmetry
- *axial symmetry,* where the problem is such that it can be modelled as a quasi-one- or two-dimensional section in some radial coordinate and a physical length coordinate
- *cyclic symmetry,* where the geometry and boundary conditions allow one part of the geometry to be modelled as a representative section of the whole geometry
- *repetitive symmetry,* where sections of the geometry repeat themselves.

Of these, mirror and axial symmetry are fairly common, but cyclic and repetitive symmetries are not. Hence, for the last two types, a significant degree of understanding is required to apply them. Further details can be found in NAFEMS (1986).

5.7 OVERVIEW OF MESH GENERATION

5.7.1 The Need for a Mesh

In Chapter 3 the finite element method was considered as a way of discretizing a structural problem so that numerical equations describing the deformation of the structure are produced. To do this requires the discretization of the volume of the structure under consideration such that a set of points, known as the nodes, is created that defines the subregions of the structure, known as elements.

The simplest meshes are for those structures that can be considered to consist of one-dimensional members such as beams and bars. These structures can be easily described by nodes and elements generated from the locations of the end points of the members. In subsequent sections of this chapter only meshes for more complex problems which are described in two and three dimensions will be considered.

5.7.2 The Constituent Parts

From the discussions in Chapter 3, it is already known that a mesh consists of subregions known as elements on which the variables are found at fixed points known as nodes. These then are the basic parts from which any mesh is built. To define any mesh the nodal coordinates must be calculated in the appropriate coordinate system, and then the elements themselves can be defined by listing the nodes that are attached to each element. This list is known as the *connectivity list*. At this time it may be that the material properties of the element also have to

be given as well as a flag to denote the type of element and its mathematical description. A variety of element types are presented in Table 3.2.

Knowing that nodes and elements need to be created, it must now be decided which elements are to be used.

5.8 ADVICE ON CHOOSING ELEMENT TYPES

5.8.1 The Range of Elements and Testing the Elements

The available elements can be listed in the following six categories:

- membrane elements (and plane stress and plane strain)
- bending elements
- general shell elements
- solid elements
- axisymmetric elements
- specialized elements like springs, gaps, rigid bars and elastic foundations.

It is not possible to present a set of universal guidelines to develop any finite element model as each structural problem and element type have their own particular features. It is not even possible to give rules for what appear in packages to be identical element types since their formulation can be different. As an example, some packages contain the six-noded triangular element described by (3.25), while others collapse a line of nodes of an eight-noded quadrilateral to form the triangle. Although the two elements are geometrically the same they behave differently when loaded.

To determine the behaviour, or performance, of elements, it is up to the analyst to perform a series of simple tests known as *patch tests*. These are implemented by seeing if a group of elements, which are usually distorted, can model constant stress and be strain-free when subjected to a rigid-body movement.

As a guide:

> Any test for element behaviour should be more complicated than the situation of a simple rectangular geometry with a constant load, since simple situations can give a false impression of the convergence characteristics for realistic problems.

Such tests are particularly important in developing an understanding of the behaviour, accuracy and limitations (e.g. 'locking' and mesh instabilities) of the various finite elements available, as well as the approximations made to represent real boundary conditions (loading and restraints) and the solution algorithms. Here the performance of any element must be established against a known solution. This must always be performed for plate and shell elements

because of the large number of formulations and the fact that the vendor's documentation is unlikely to describe fully the formulation and features of its available elements. It is difficult to make a good judgement about which element to use unless simple tests are conducted. By word of warning, most of the research effort has focused on element formulations and convergence criteria, but the test cases for evaluation are usually small-sized academic cases (including patch tests) rather than actual problems. So the demonstrations of adequacy in the research cases may not be very helpful in the practical cases where an analysis must be done. NAFEMS have developed a series of benchmarks which are a valuable aid.

Current practice indicates the following:

- Quadratic elements, be they membrane or solid elements, give the best compromise between accuracy and efficiency for general use.
- Even though a finite element package has the linear tetrahedron solid element (see Table 3.2) in its library it should not be used because it has been condemned by the analyst community.
- When modelling a structural problem that can be classified as having bending (with or without membrane) deformation and the geometry is either flat or curved, then the preferred choice of element is always the general shell element based on the isoparametric Mindlin theory.
- Curved surfaces should not be modelled using flat elements as the discontinuity at element boundaries introduces significant error.

5.8.2 Using a Hierarchy of Elements

When approaching a new type of problem the experienced analyst should perform a parametric study to investigate combinations of structural and element behaviour related to the problem. For instance such studies can be used to determine the effect of die-away lengths.

Another example is the use of an approximate two-dimensional model before building a three-dimensional model. This undoubtably saves resources as the two-dimensional model provides invaluable information for the three-dimensional mesh specification in terms of the mesh density.

As a guide:

> Analysts should develop a model using a step-by-step approach. This means that they should start with a simple approximation, say a beam model, and make it more precise as the finite element modelling progresses. Never tackle a real problem directly as this is likely to be time consuming and wasteful of resources. Remember, the more results that are generated the more effort that will be necessary to check that they are reliable and relevant.

Equally, results from one model can be used to provide boundary information for another. An example of this is that the results from a general shell model may be used to feed information to a model generated with three-dimensional solid elements of a local region of interest. This is a process known as *substructure modelling*. In this case it is necessary to plan the shell mesh so that a local group of elements will provide a suitable set of boundary information for the local solid model.

5.8.3 Restricting the Dimensions of a Problem

By using plate or shell elements instead of solid elements not only is the computing effort reduced, but also the accuracy may be improved. As a guide:

> Avoid the use of solid elements to model a problem where the length in one of the spatial dimensions, for example the material thickness, is much less than the lengths in the other two dimensions.

This reduction in accuracy with solid elements is due to ill-conditioning. To achieve acceptable aspect ratios for the elements the mesh will require many elements and Eq. (1.2) will therefore have a large number of degrees of freedom. NAFEMS (1986) illustrates this feature using the example of a simple cylinder whose radius is 10 times the wall thickness. Here the choice of element can be either axisymmetric thin shell, axisymmetric thick shell, general thin shell or general solid.

As an aside, note that with computing power growing rapidly there may come a time in the not too distant future when the use of solid elements in this situation does become realistic.

5.8.4 Plate and Shell Elements

Plate and shell elements have historically been the most difficult to use in terms of achieving reliable and cost-effective solutions. At this time, there is no particular element that is broadly acceptable within the analysis community.

In particular these elements in a static analysis do not give an acceptable solution if the displacement of the nodes normal to the surface of the material is greater than the thickness of the material.

The following guidelines may also be of use:

- If the structural problem consists of flat panels of material connected into open or closed sections (e.g. I-sections, box sections or stiffened skins) flat plate elements are inappropriate as the in-plane effects are generally important.

- When solving a problem by plate or shell elements never trust the results without making considerable effort to verify the correctness of the model as well as the solver algorithms and solutions.
- Depending on the properties of the material there will be a thickness value of a shell structure above which the model must have thick shell elements in preference to thin shell elements.
- When a model has thick plate elements or their shell element counterpart then the presence of 'locking' should be easy to recognize because the results will be low. Refining the mesh such that the aspect ratio of the length of an element side to the thickness is reduced should eradicate the detrimental effect of locking.
- The plate or shell theory is not appropriate in regions where panels join or curved surfaces make sharp radius turns.
- When using beam off-set elements and shell elements to model a stiffened skin structure there may be a significant error associated with the coupling of beam and shell elements, especially for coarse meshes.
- Avoid distorting the shape of plate and shell elements from their parent shape as they are highly sensitive to any type of distortion.

5.8.5 The Role of Compatibility

If elements are compatible internally and across their boundaries then, as the mesh is refined, the solution will convergence monotonically to the exact solution of the finite element method. Note that the exact elastic solution is never achieved because the number of nodes, or degrees of freedom, must be optimized to balance the errors due to discretization and round-off.

Remember that plate elements and general shell elements do not satisfy compatibility completely. In fact, complete compatibility requires the following:

- Elements must have the same order, although one can mix three-sided and four-sided elements.
- There must be connection between the corner nodes of neighbouring elements and, if present, continuity between the edge nodes of adjacent elements.

Special transition elements allow the element order to change in a mesh, but transition elements must not be used in regions where accurate stress values are required.

5.8.6 Elements to Model Contact

In this situation the following guidelines may be of use:

- Before developing a three-dimensional model for a problem with contact between different parts, check that the package has three-dimensional contact algorithms as these are a current topic of active research.

- If the finite element package has gap or interface elements to model contact between different parts of a structure, it is often a requirement to have adjacent nodes on the surfaces of the separate parts.
- If the package has elements to model the penetration of one surface by the surface of another part and push it back (known as sideline or contact patches) there is not usually the requirement to have adjacent nodes on the surfaces of the separate parts. Note that impact problems are handled in a different way.

5.9 DETERMINING A STRATEGY FOR DISTRIBUTING THE ELEMENTS

Once the list of useful elements has been decided, the analyst must determine how the elements should be distributed. At this stage it is probably sufficient to consider those areas that require large numbers of elements and those areas that can be modelled coarsely. Clearly, the choice of the mesh density is dictated by the element type that is to be used and the expected stress distribution throughout the structure. In the following sections some simple guidelines will be given.

5.9.1 Coarse or Fine Distributions

A coarse mesh may be used in cases where limiting the deflection is the principal design criterion for a complex three-dimensional solid structure where the analysis is global. Here, the geometry may be approximated by a coarse mesh of flat-faced solid elements.

Fine meshes are required in the following examples:

- When accurate stresses are necessary for a detailed failure analysis. Here, curved solid elements with minimum distortion may be used which fit the surface geometry as closely as is practical. In this case, the region of the structure that is of particular interest needs to be meshed very finely.
- When the effect of joints is modelled. Here the effect is often underestimated, but it may have an appreciable effect on the global, as well as the local, behaviour. In fact, it is much easier to devise a mesh for a complicated smooth three-dimensional structure than it is to model a joint accurately within a simple framework.

One of the main advantages of the finite element method over other numerical methods is that it allows a variable mesh density. Despite this, as a guide:

> The best results will always be obtained if the size of the elements in the mesh is constant and their shape is that of the parent element.

Often this is not possible and so the following guidelines may be of use:

- The volume of adjacent elements with identical material properties should not differ by more than two to four times. If there is an abrupt change of material properties the adjacent element volumes should be chosen such that the stiffness terms in their element stiffness matrices are similar.
- Avoid having elements of significantly different sizes in a model as this provokes ill-conditioning. As a rule of thumb the ratio of maximum to minimum element volume within a mesh should not exceed 30.
- The more gradual that any transition of material properties, loading, constraints or geometry can be made then the better the results will be. This is because the finite element method gives, in general, a smeared average solution with accurate results at only a limited number of points within each element. These points are not necessarily the nodes or the Gauss points.
- To minimize the problem of ill-conditioning when the element includes in-plane membrane and bending deformations it is advisable to have a mesh with a higher level of refinement than would be adequate if the element had either of these deformation components alone.

Distortion of elements can also cause problems as numerical errors are introduced. This is often the situation when the meshes and geometries are irregular. A list of the types of distortion is given in Chapter 3 and includes non-unity aspect ratio, angular distortion, volumetric distortion, warpage angle distortion and edge-node position distortion.

Packages check for distortion by determining the value of the Jacobian. It is often the analyst's responsibility to carry out checks to determine the maximum allowable distortions for any element.

5.9.2 Modelling Discontinuities

High rates of change in stress (and strain) exist wherever there is any type of discontinuity. For example, there can be discontinuities due to abrupt changes in such things as geometry, loading, material properties and supports.

As the mathematical formulation of elements tends to smear out the variation of displacement over an element, this has serious consequences when discontinuities are to be modelled. The finite element method is most accurate for continuum problems where the geometry, material properties and boundary conditions (loadings and constraints) all change in a smooth manner. If discontinuities of any form are present they are effectively smeared out and the finite element modelling is not precise. Refinement of the mesh helps to reduce problems, but doing this is too costly in three-dimensional problems, as to refine the mesh in all directions leads to an increased analysis cost which is roughly proportional to the cube of the number of elements.

Multi-point constraints can be used to allow abrupt changes in element geometry. This can be useful when mixing element types (e.g. two-dimensional membrane continuum elements and one-dimensional beam elements or three-

dimensional solid continuum elements and general shell elements) as the analyst can ensure that the connecting degrees of freedom are matched by using multi-point constraint equations.

Once a choice of element or elements has been made and a distribution strategy decided, then the form of the mesh structure must be determined. By arranging the elements in different topologies, various forms of mesh can be built; this will be discussed in the next section.

5.10 TYPES OF MESH STRUCTURE

5.10.1 Regular and Irregular Mesh Structures

Now that the constituent parts of a mesh are known, the arrangement of these parts through the domain can be considered. This arrangement is known as the *form* or *topology* of the mesh. When using the finite element method, the points are the nodes of the set of elements used to split up the material volume and the elements can be arranged in any way, providing that the faces of the elements are positioned correctly. This means that, to ensure compatibility of the mesh, the edges of two-dimensional and the faces of three-dimensional elements which are touching must be in contact, with edge exactly matching edge or face exactly matching face and with node matching node. Clearly, one-dimensional elements are only placed from node to node. This less restrictive form is allowed as the calculation on any one element requires information from that element alone. The interaction between the elements takes place when the element equations are added together to form the global equations as discussed in Sec. 3.9.

From this it can be seen that there are two ways in which the mesh structure can be arranged. The first is a *regular form* or *topology*, where the nodes of the mesh can be imagined as a grid of points placed in a regular way throughout a cuboid. These nodes can then be stretched to fit a given geometry and this is shown in Fig. 5.3. Note that when the mesh is stretched the connections between the nodes do not change. The stretching takes place as if the mesh were made of rubber, and the topology of the mesh remains the same. Consequently, if any node in the mesh is considered it will be connected to the same neighbouring nodes both before and after the stretching process. Sometimes these meshes are called *structured meshes* as they have a well-defined structure, or *mapped meshes* as they can be seen as a cuboid mesh that has been mapped onto some other geometry. When considering these meshes it is useful to think of a *local coordinate system* within the mesh. This enables the orientation of the cells relative to each other to be determined, and so before the mesh is transformed the axes of this system are the edges of the cuboid. Once the transformation into the actual coordinate system, the *global coordinate system*, is carried out, then the local coordinate system axes become dependent on the position within the mesh. This is shown in Fig. 5.3.

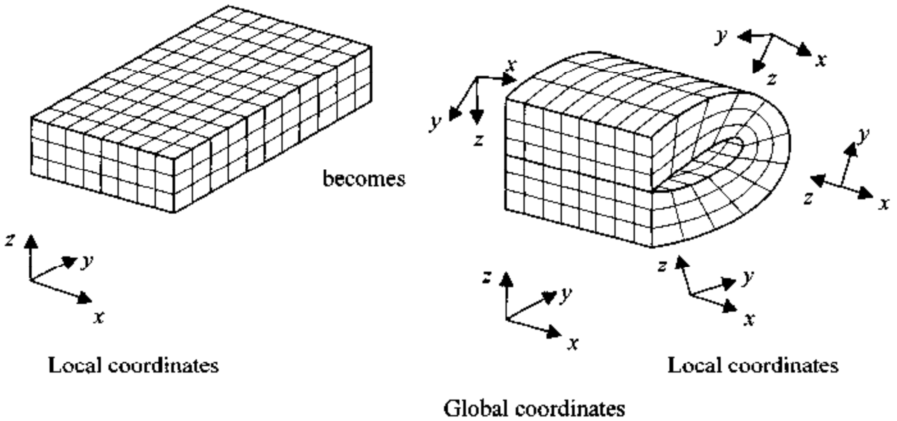


Figure 5.3 Transformation of a mesh with a regular structure.

The second arrangement is an *irregular form* or *topology*, where the nodes fill the space to be considered but are not connected with a regular topology. Figure 5.4 shows a two-dimensional example of this type of mesh formed with triangular elements. Note that the element faces do not overlap and that elements are labelled by numbers and nodes by letters. It can be seen from

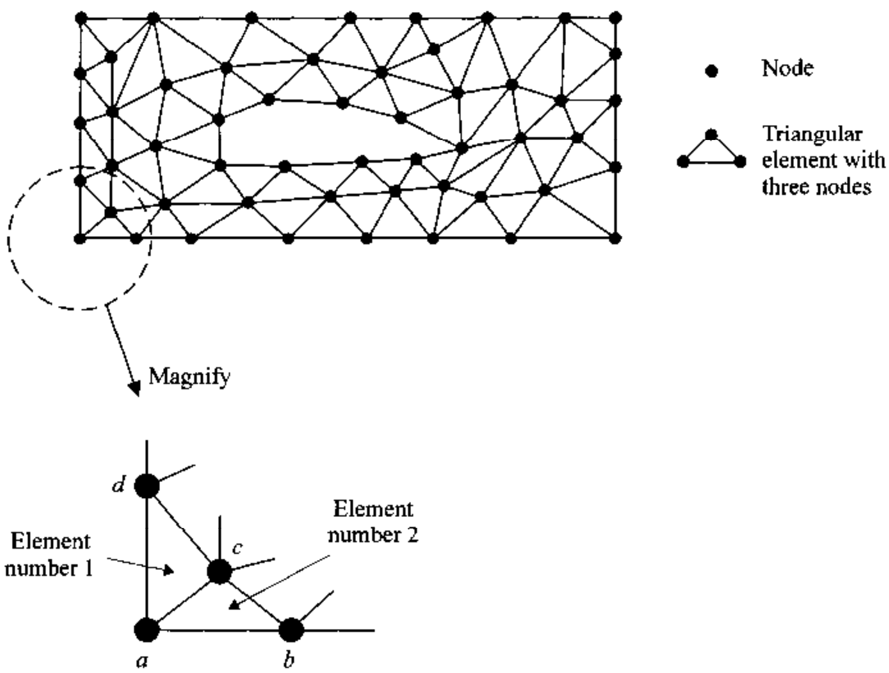


Figure 5.4 A mesh with an irregular structure.

the magnified section of the mesh that element number 1 has the three nodes labelled *a*, *c* and *d* at its corners, and that element number 2 has the nodes labelled *a*, *b* and *c* at its corners. The fact that any particular node is attached to an element cannot be known from the form of the mesh, and so a numerical table must exist that describes the arrangement of the mesh by listing which nodes are attached to each element. This contrasts with the regularly structured mesh where a knowledge of the location of an element within the mesh enables the labels of the points at its corners to be found implicitly. A mesh with an irregular structure is often referred to as an *unstructured mesh* or a *free mesh*.

Many structures that are of interest to engineers have complex geometries. With some ingenuity on the part of the analyst, it is possible to fit a mesh with a regular form to some of these geometries, but with many geometries this is not possible. This is where meshes with an irregular form can be used to great advantage, as these can be used to describe the most complex of geometries since there is no restriction on the form of the mesh. This can make the mesh generation process much easier and in some cases it is a prerequisite for producing a simulation. Another advantage of using irregularly structured meshes is that they can be created by automatic mesh generation algorithms, some of which are described in Sec. 5.11.3. These algorithms generate meshes that are unstructured using elements such as plane triangular elements and tetrahedral elements.

5.10.2 Determining the Mesh Form

Having chosen the element types and having made sure that the computer model of the geometry is complete, the next step is to decide the type of mesh form to be used. Often this is done by trying to find topological rectangles or bricks in the geometry in an attempt to see whether a regular mesh can be used. If this is not possible then an irregular mesh structure is appropriate.

Having decided on the mesh form, a mesh layout can be determined and an estimate made of the number of elements required. To do this requires considerable user experience; both the layout and number of elements depend on the strain (and stress) gradients that occur within the structure as it is deformed, as was seen in Chapter 3. In particular, the mesh layout depends on five things:

- the geometry of the structure
- the type of analysis, i.e. static, dynamic, thermal or nonlinear
- the boundary conditions
- the loadings
- the required results.

Remember that when constructing a mesh the elements must be arranged in such a way that the arrangement does not significantly affect the results. For example, with a regular mesh of triangular elements for membrane action it is

good practice to have the direction of the sloping side alternating. The practice of having all the sloping sides in one direction is to be avoided.

5.11 BUILDING MESHES

5.11.1 Building Simple Meshes with a Regular Form

Many problems can be solved by using a mesh that has a simple regular form. One common way of producing a regular mesh is to use the hierarchy of entities as shown in Fig. 5.5, or something similar. In this figure the hierarchy for four-noded two-dimensional elements or eight-noded three-dimensional elements is considered. At the bottom of the hierarchy is the most basic geometrical entity which is a point, several of which can be linked to form lines (from two points), arcs (from three points) or splines (from four points or more). By combining adjoining lines, arcs and splines the third-level entity, the edge, can be created. If four edges form a closed loop they can be seen to be the boundaries of a surface and six of these surfaces can be used to bound a volume. This set of relationships is determined by the form of the elements being considered. Once the surfaces for a two-dimensional problem or the volumes for a three-dimensional problem are defined, the elements can be formed. This is done by mapping the surfaces into a square, and by mapping the volumes into a cube. These squares and cubes are used to define a local coordinate system in which the parent elements can be created before these are transformed back to the global coordinate system in which the real elements are defined. Here, the techniques of transformation that are used are similar to those used in forming the isoparametric elements discussed in Sec. 3.7.

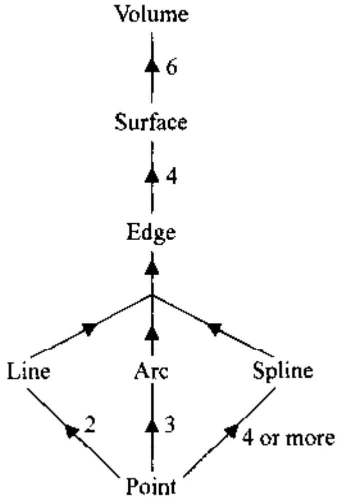


Figure 5.5 A hierarchy of entities.

Another way of building regular meshes in a topologically rectangular domain is to use the techniques of Thompson, Warsi and Mastin (1982), where partial differential equations are developed that describe the nodal positions within the domain. These techniques generate meshes which are particularly smooth in their variation from element to element.

5.11.2 Using Commercial Pre-processors

Commercial mesh-generation packages have been around for some time now aimed at the finite element structural analysis market. These packages usually have the following components:

- A geometry creation routine, where one-, two- or three-dimensional geometrical data can be created in the form of points, lines, arcs, splines and, sometimes, surfaces. An interface to extract similar data from CAD systems is a common feature as well.
- A domain definition routine. This allows the creation of surfaces, in two dimensions, or volumes, in three dimensions. These domains are the spatial entities on which the mesh is to be built.
- A mapped-mesh generation routine. This enables a mesh with a regular structure to be created within the domains. These domains must be topologically consistent with the element type being used. For example, if four-noded quadrilaterals are being used to mesh a two-dimensional domain, then a four-edged domain must be used.
- A free-mesh generation routine. This enables a mesh without a regular structure to be created within the domain. In this case there is, in principle, no restriction on the form of the domains, and so they can be either surfaces bounded by any number of edges (for two-dimensional problems) or volumes enclosed by a set of these surfaces (for three-dimensional problems).

When using commercial mesh-generation software, hierarchies such as that shown in Fig. 5.5 are used. Usually, this does not cause a problem, but there is one area where errors in the modelling of a geometry can occur. Coons patches are an obvious choice for defining the geometry of a surface within the hierarchy, as the edges are used to define the surface. As was discussed in Sec. 5.4, such a representation of a surface may not be adequate if the patch is too large for the curvature of the surface. One way of overcoming this problem is to define smaller surfaces, but this involves much more work on the part of the analyst. Another way is to use more accurate surface descriptions, say Bézier surfaces or NURBS, derived from a CAD model of an object. Many commercial finite element pre-processors can read these more accurate surfaces from the database of a CAD system. Then, a set of edges can be used to define a Coons patch surface. Once this has been done, the user can tell the pre-processor to calculate the mesh points on this surface by first calculating the coordinates of the points on the Coons patch and then recalculating the coordinates so that they are positioned on the exact surface.

5.11.3 Automatic Mesh Generation Algorithms

For simple geometries it is easy to see how a mesh can be built, but when the geometry becomes more complicated the meshing process is more difficult. Several techniques have been developed that can take complex two- and three-dimensional geometries and then automatically produce a mesh that models the geometry. Typically, the mesh will have an irregular structure. As was stated, generation is a costly part of the analysis process because of the large amount of human effort that can be required to build the mesh for a complex geometry. Any savings in the time taken to build a mesh reduce the cost of an analysis, and so these automatic mesh generation techniques are being actively researched (George, 1991).

The first method that will be discussed is *Delaunay triangulation* (Cavendish, Field and Frey, 1985; Holmes and Lanson, 1986; Watson, 1981). Figure 5.6 shows this algorithm at work for a two-dimensional case where triangular elements are to be created. The algorithm is easily extended to three dimensions where tetrahedral elements are formed. Figure 5.6(a) shows that the basic technique is started by producing nodes on the boundary of the domain and nodes inside the boundary. In this case there are 12 nodes on a square boundary and one node inside the domain. To ensure that the final triangulated mesh has no gaps in it, three extra nodes are then created that define a super-triangle. From Fig. 5.6(b) it can be seen that these extra nodes have to be placed so that they define a super-triangle that encloses all of the original nodes of the problem. This super-triangle is taken to be the first element and then one of the original nodes is used to split this element into three new elements (Fig. 5.6c). Now an iterative element creation procedure can begin. One by one each of the remaining nodes is considered and the mesh modified. To do this a circle is created for each element such that it passes through each of the three nodes of that element. Figure 5.6(d) shows the circles of the elements and node 2 will be considered. This node lies outside two of the circles and inside the other. The triangulation algorithm states that if a node lies inside a circle then the element to which the circle is attached should be deleted. Once all the necessary elements have been deleted, new elements can be created that include the node being considered. This is shown in Fig. 5.6(e) where the lower element of Fig. 5.6(c) has been deleted and three new elements have been created which are joined at node 2. Then another node is considered and the process continues. Eventually, a final mesh is created such as that in Fig. 5.6(f). This can be modified so that only the original domain, in this case the square, is modelled. This is done by deleting all the elements which are attached to the nodes that formed the super-triangle. Finally, the shape of the remaining elements is checked and, where necessary, the mesh is modified using face or edge swapping and node insertion to give elements as near to equilateral triangles as possible. At the end of this process the mesh does not have elements with a very distorted shape as these elements lead to numerical errors when calculating the element equations as discussed in Sec. 3.7.6.

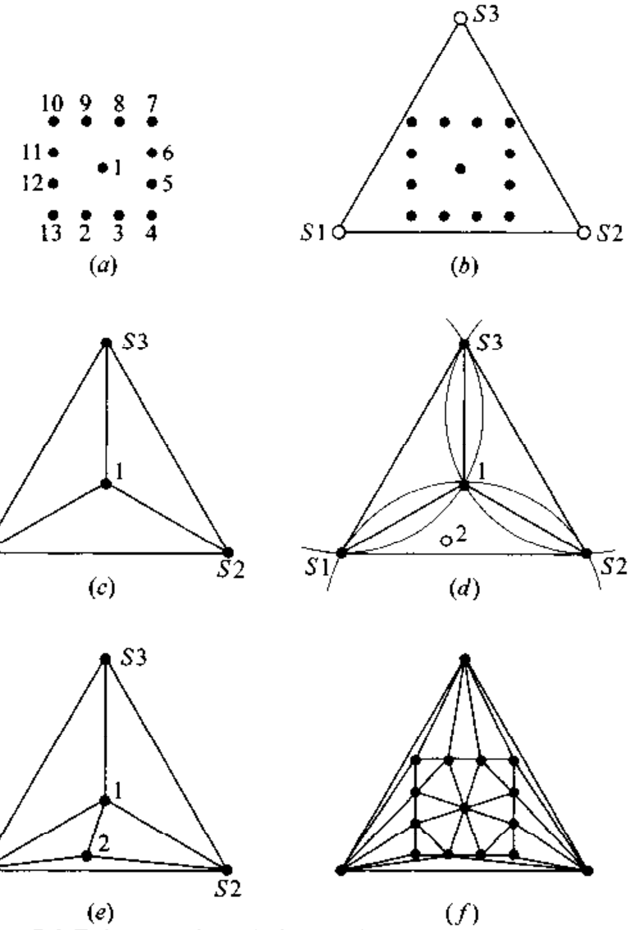
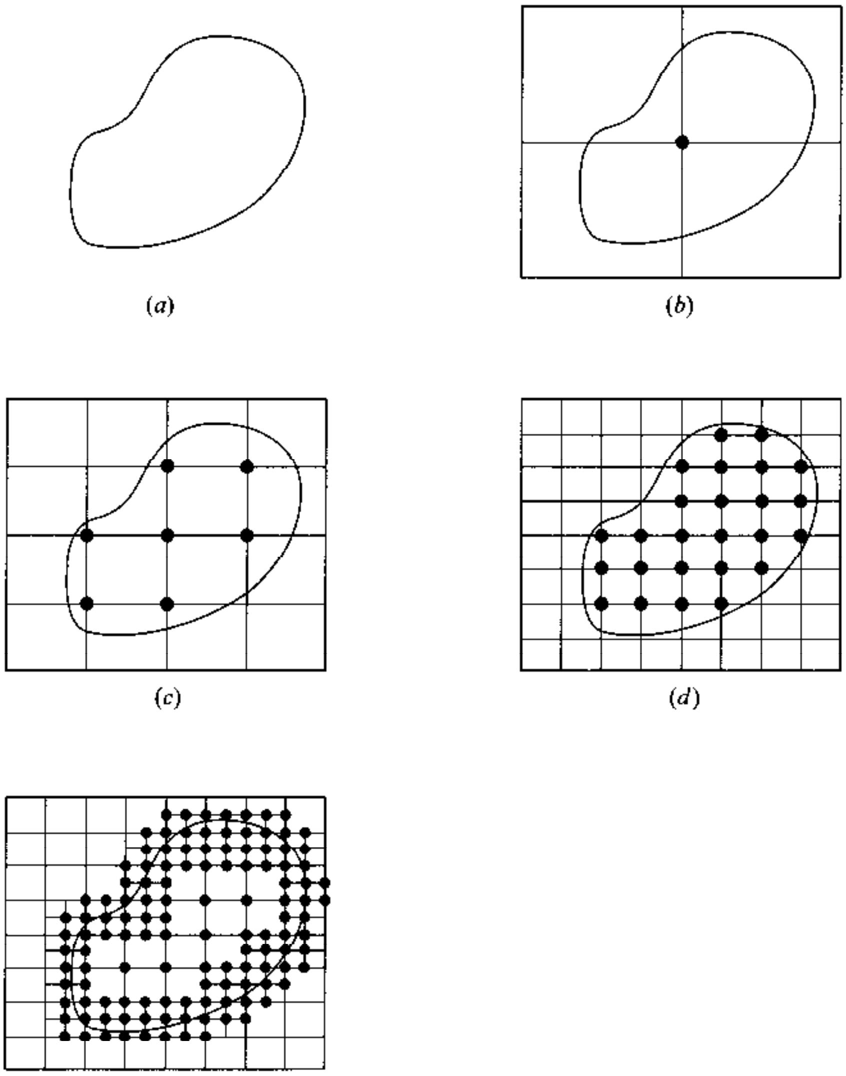


Figure 5.6 Delaunay triangulation: (*a*) boundary and internal nodes, (*b*) addition of super-triangle, (*c*) the first three elements, (*d*) forming circumcircles, (*e*) element deletion and recreation, (*f*) the final mesh.

The second method is based on the use of the *quadtree* and *octree* methods (Yerry and Shephard, 1984; Cheng, Finnigan, Hathaway *et al.*, 1988). These methods take a domain and place it inside four squares, if it is a two-dimensional problem, or eight cubes, for a three-dimensional problem. These are then subdivided until the required definition is acquired. Hence the name 'quadtree' refers to the structure of the elements in two dimensions and 'octree' refers to the three-dimensional method. Figure 5.7(a) shows an example of a two-dimensional domain that is to be meshed. Four squares are placed over the domain, as shown in Fig. 5.7(b), and a node is created where the squares are joined inside the domain. Each square can then be subdivided into four more squares and more internal nodes created. Two further subdivisions are shown in parts (c) and (d).



(a) (b) (c) (d) (e)

Figure 5.7 The quadtree method: (*a*) domain to be meshed, (*b*) first overlay, (*c*) second overlay, (*d*) third overlay, (*e*) selective subdivision at boundaries.

Once the element size for the bulk of the mesh is small enough, only the elements that cover the domain boundary are subdivided. This selective subdivision is shown in Fig. 5.7(e), and it can be continued as required. Note, however, that it leaves a mesh that is a stepped representation of the domain. To overcome this the external nodes are moved so that they are on the surface of the domain. Finally, triangular elements can be created to link all the nodes.

A third method that is increasingly popular is the *advancing front method.* This is described in detail by George (1991). Essentially, the method requires a mesh on all of the surfaces of the boundaries of the structure. This consists of line elements in two dimensions or triangular elements in three dimensions. These form a so-called *initial front* on which new elements, triangles in two dimensions or tetrahedra in three dimensions, are built. Where possible, equilateral triangles are used either on their own or to build the tetrahedra, so ensuring that the mesh has optimal shape properties. Once the first set of elements has been added the new front is established by removing the old face elements from the front list and adding the new faces as appropriate. Elements are added in this way until the whole volume is filled and the front has no members. Figure 5.8 shows the process in action.

Specialized software is available to perform mesh generation using forms of Delaunay triangulation, quadtree/octree and advancing front methods, but commercial finite element mesh generation software can also be used to generate a mesh with an irregular structure in an automatic way. This is often done by meshing the surface of the domain, using triangular elements. Then the volumes that have been defined by the surfaces can be meshed using tetrahedral elements formed from the elements on the surface. At first sight this might appear to restrict such free-mesh generation methods to using only tetrahedra. These can, however, be easily converted to eight-noded brick elements as shown in Fig. 5.9. There a single tetrahedral element is taken and new nodes formed at each of the mid-sides of the element edges, at the centroids of each face of the element and at the centroid of the whole element. These can then be joined as shown to produce four eight-noded brick elements.

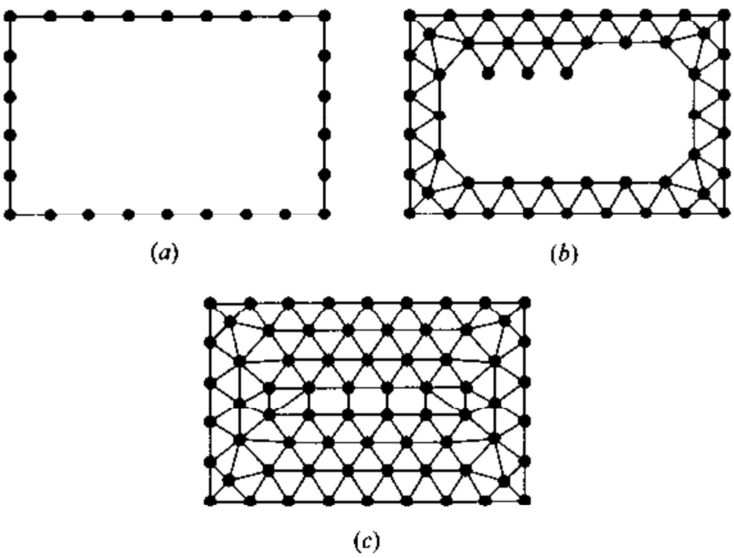


Figure 5.8 Mesh generation using the advancing front method: (*a*) the domain boundary (first front), (*b*) front advancing into domain, (*c*) final mesh.

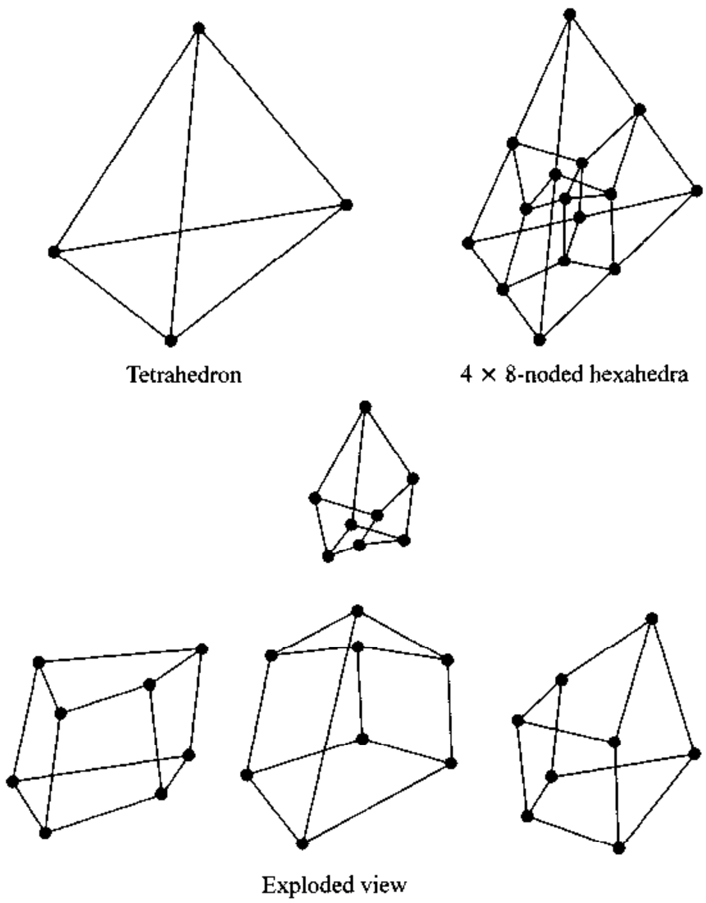


Figure 5.9 Transforming a tetrahedral element into brick elements.

Note that a three-dimensional mesh is very much more difficult to visualize when it is generated using an automatic routine. In this case mesh checking becomes a more important stage in the model development, as much of the model and many of the element edges and nodes are hidden from view in any graphical display.

5.12 ADAPTING A MESH TO IMPROVE A SOLUTION

5.12.1 Refinement and Enrichment

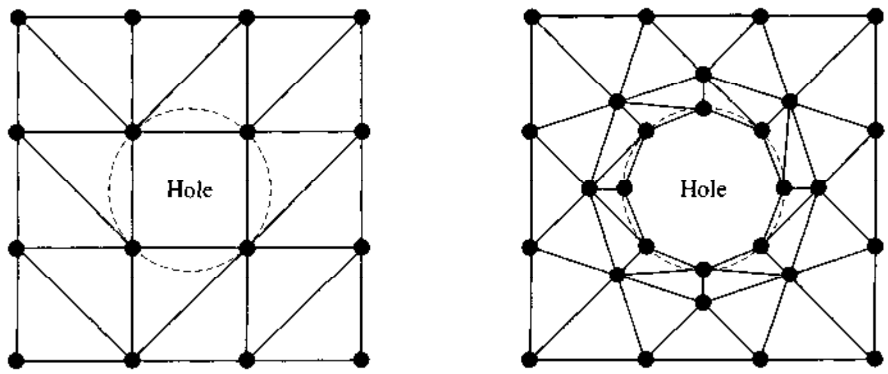
Once a mesh has been built it is possible to modify it in such a way that the solution that is produced on the modified mesh is a better one. This modification can take place before a solution to the structural problem is found, using an

analyst's intuition to improve the mesh. Equally, it can take place after a solution is found, using either intuition again or some automatic means to refine the mesh such that a new and improved solution can be generated.

On many occasions, the solution will be good over most of the model but will need more refinement in one or two regions. On other occasions the model may be highly complex with many components such that refinements in all the components soon become impractical. This situation calls for the ability to model a subregion separately while including the influence from the overall structure.

Mesh modification techniques can be applied after a solution has been produced on an initial mesh. These techniques are used to modify the mesh in the light of the results achieved on it and so the dependence of the quality of the results on the user's experience is reduced. These modification procedures require that an initial analysis is made using a crude but realistic mesh of the domain. From the results of this initial analysis the mesh is recreated such that the density of the mesh points is greatest in regions of the domain where the strain gradients change rapidly or where the error in the numerical equations is found to be large (Zienkiewicz and Taylor, 1989). The mesh is said to be adapted to take account of the results generated. Several types of mesh modification are commonly used:

- *Mesh enrichment*, where additional points are placed within the domain at the locations where they are needed as shown in Fig. 5.10. In this figure a mesh is required for a model of a plate which has a hole in it, leading to a stress concentration when it is loaded in the plane. The original mesh of triangles has a regular spacing but the enriched mesh has additional nodes and elements in it so that there are more elements near the stress concentration. This technique is usually applied to meshes that consist of triangular elements in two dimensions and tetrahedral elements in three dimensions. Such meshes allow additional points to be created in the mesh and then the Delaunay triangulation method, or similar methods, can be used to create a new set of elements.



(*a*) (*b*)

Figure 5.10 Mesh enrichment: (*a*) original mesh, (*b*) enriched mesh

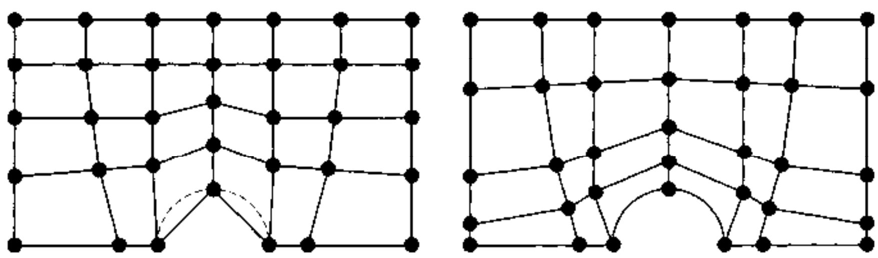
- *Mesh adaption,* where the topology of the mesh stays the same but the mesh points are moved so that the density of points increases where required as shown in Fig. 5.11. Here, a stress concentration is again modelled. Note that the number of nodes and elements remains the same in the adapted mesh. Only the node positions are changed. This movement of the points can be brought about by using modified forms of the partial differential equations that are used in some grid generation methods as was discussed in Sec. 5.11.1.
- *h-refinement,* where a mesh is refined by systematically subdividing elements into more elements of the same order. For example, a quadrilateral element could be converted into four quadrilateral elements placed within the same area. On each h-refinement the previous mesh form can be seen within the new refined mesh form.
- *p-refinement,* where the elements are modified to be elements covering the same domain but with an increased order of polynomial interpolation. For example, linear elements could be converted to quadratic elements then cubic elements and so on.
- *hp-refinement,* where both of the above techniques are used together.

By using these smoothing or adaption techniques the accuracy of the solution can be increased, but there is a penalty in that extra computational effort is required. In some commercial packages adaptive meshing is implemented as a way of achieving automatic convergence of finite elements through mesh refinement or remeshing a model based on error estimates.

5.12.2 Reducing the Bandwidth

As mentioned in the Sec. 3.9, the use of direct solvers for many structural finite element solutions leads to a solution time proportional to the number of variables times the half-bandwidth squared. For the sparse matrices that are produced the half-bandwidth is a function of the way in which the nodes of the mesh are numbered.

For a simple mesh such as that shown in Fig. 5.12, it is easy to see that different ways of numbering the nodes can be chosen. The mesh on the left has



(*a*) (*b*)

Figure 5.11 Mesh adaption: (*a*) original mesh, (*b*) adapted mesh.

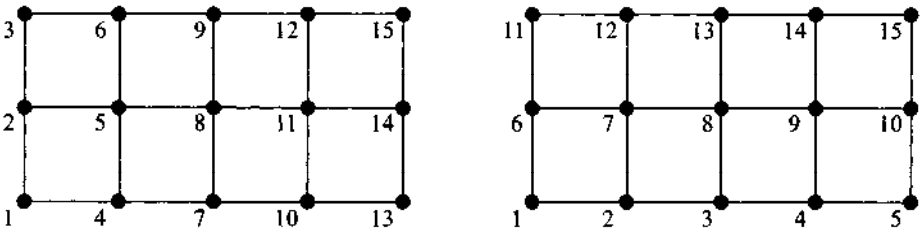


Figure 5.12 Effect of node numbering on half-bandwidth. All elements are four-noded quadrilaterals.

a bandwidth of four while that on the right has a bandwidth of six. Clearly, the analyst can influence the size of the bandwidth during mesh preparation and so it is common for manual methods of bandwidth reduction to be built into mesh creation strategies. These are discussed by NAFEMS (1986) and Desai and Abel (1972).

For complex meshes, however, manual intervention in the node numbering problem for minimum bandwidth is virtually impossible. Hinton and Owen (1979) discuss this and show an example of an automatic procedure to reduce the bandwidth of a mesh. This is based on the method of Cuthill and McKee (1969) which utilizes the quantity known as the degree of a node, i.e. the number of nodes connected to a given node, to reduce the bandwidth. The method is illustrated in Fig. 5.13 by a flowchart. Note that a start node must be chosen by the user and is assigned as node 1. Then all nodes attached to this node are numbered in sequence 2, 3, 4 and so on in increasing order of degree. Then the nodes attached to node 2 are renumbered in the same way. This is repeated for all nodes in the mesh. There is no guarantee that this will be an optimal numbering and so it is common for the algorithm to be repeated with a variety of nodes being taken as the first.

Figure 5.14(a) on page 139 shows a mesh of triangular and quadrilateral linear elements together with the degrees of each node. Two different solutions obtained using the Cuthill–McKee algorithm are also shown in parts (b) and (c). Note that the bandwidth is increased from 7 in part (b) to 8 in part (c).

Clearly, several attempts have to be made as different start positions will give different values for the bandwidth. The Gibbs, Poole and Stockmeyer (1976) algorithm addresses this problem by finding a start node with the minimum degree on the boundary of the domain and finding an optimum path from the start node. This is also discussed in George (1991) together with a variety of other methods.

5.13 CHECKING A MESH

Once a mesh has been built it is difficult to check the data for errors. To assist in this process various techniques can be used and most are built in to commercial pre- and post-processors.

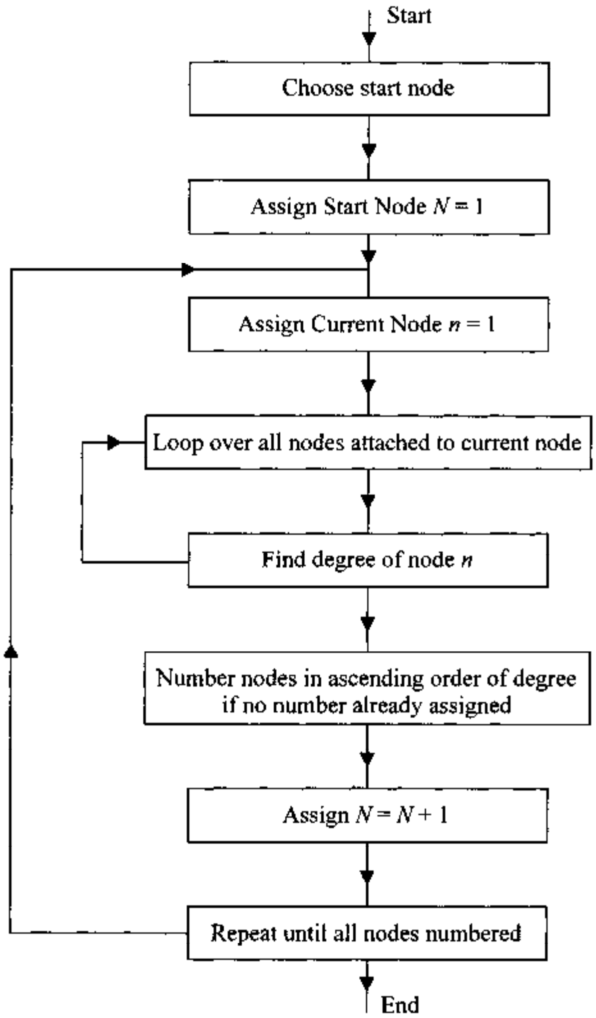


Figure 5.13 A node renumbering algorithm.

The following checks are common:

- *Finding coincident nodes* A tolerance is given and the positions of all nodes are checked against all other nodes for coincidence. Any sets of nodes that are listed as coincident can usually be merged together by the software and the appropriate changes made to the connectivity list automatically.
- *Free-face checks* The faces of the elements that are only connected to one element are displayed. This is done using the algorithm given in Chapter 6. All free-faces should be on the true boundary of the mesh and so this method shows any holes in the mesh that have been created inadvertently.

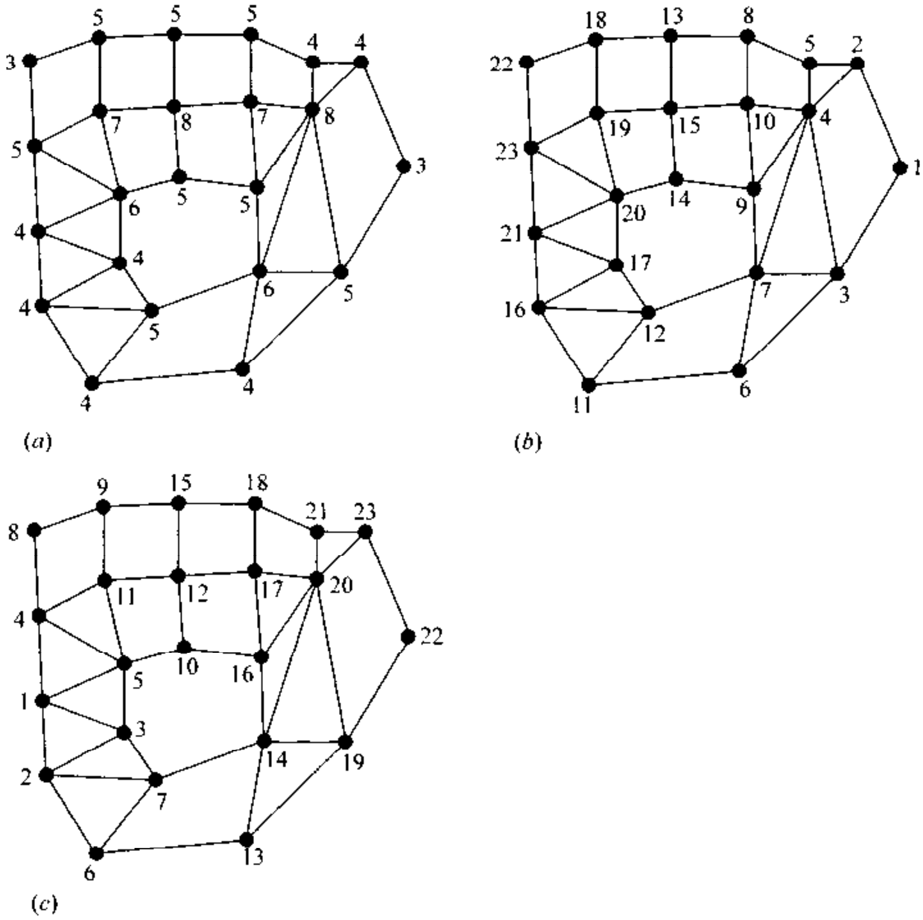


Figure 5.14 Node renumbering with the Cuthill–McKee algorithm: (*a*) the degree of the nodes, (*b*) case 1—bandwidth of 7, (*c*) case 2—bandwidth of 8.

- *Free-edge checks* This is similar to a free-face check but the free edges of the mesh are displayed. Again holes in the mesh, created accidentally, are highlighted.
- *Exploded mesh plots* The elements are drawn on the computer screen in a shrunken format. This aids visualization of the mesh.

6

OBTAINING AND ANALYSING THE RESULTS

This chapter contains material previously published in Shaw (1992), by kind permission of Prentice Hall.

Once a mesh has been built to describe the domain occupied by the structure, the rest of the computer model can be built. It is only at this stage that the description of the physical problem generated in the initial stage of the analysis process can be related to the computational geometry described by the mesh of nodes and elements. For each element, its material properties must be defined together with the boundary conditions on the faces of the elements, or at the nodes, which form the exterior of the mesh. The first few sections of this chapter will look at how this can be achieved for the case of a linear static analysis. Be aware, however, that for more complex problems, such as those discussed in Chapter 10 where optimization, dynamics, nonlinearity, time dependence and thermal effects are involved, much more information must be provided at this stage.

Once the computer model is complete, the solver can be run to obtain the results of the simulation. While this should, in theory, be a straightforward process for a linear static analysis, things can and do go wrong. The middle sections of this chapter look at how the solver is run and the troubleshooting that is required to ensure that results are produced. These results will be a numerical solution to the governing equations, produced with the appropriate boundary conditions on a mesh that approximates the geometry of the problem. Hence, the solution is strictly a solution of the numerical problem and not of the physical problem, any differences between these two being due to such things as an inadequate mesh or an approximate boundary condition specification.

When the numerical solution is obtained it is necessary to determine whether or not it bears some relationship to the physical reality. If it is likely that it does,

then the required technical information can be extracted from the results with some confidence. Consequently, the final sections of this chapter look at what forms the results of a simulation, how computer graphics can be used to start the evaluation of the results, how the solution can be checked to see if it is likely to be reliable, how the model can be refined so that the required data can be obtained from the results and at failure criteria.

The issues discussed in this chapter will be amplified by the discussion of the examples of finite element modelling in Chapters 7–9.

6.1 SPECIFYING MATERIAL PROPERTIES

As was seen in Sec. 2.2.6 it is not necessarily a straightforward task to define precisely the material properties and, frequently, they must be approximated when compiling the model data for an analysis. Typical property data in Table 2.1 shows that there is a degree of variation of the Young's modulus and strengths for most engineering materials. It is therefore the responsibility of the analyst to choose the appropriate value for the properties that ensures the analysis provides results for a conservative and safe structural design.

Note that data in tables, such as Table 2.1, are usually quoted as being valid at room temperature (20 °C). Hence, if the operating temperature is not room temperature, a check needs to be made on how much a property will vary between the temperatures. Furthermore, if there is a temperature variation throughout a body which is made of notionally isotropic material then the material must be considered as nonhomogeneous as the material properties will be dependent on both position and temperature. Many analysis packages have the facility to model the variation of a property with temperature and will assign a constant 'smeared' property value to each element in the mesh.

There are other occasions when the analyst is required to model real material behaviour using a smeared equivalent material property. One important application of this approach is in the analysis of a composite material that consists of two or more distinct solid phases with one of the phases having constant shape and size and being distributed uniformly throughout the second phase. There are in the literature standard techniques for calculating equivalent material properties from knowledge of the constituent properties and the distribution and volume fraction of the dispersed phase. For this approach to work the dimensions of the composite material in the structure must be greater than 20 times the smallest dimension of the inclusions. This is the situation, for example, with fibre-reinforced composite materials that are increasingly preferred to conventional materials in major structural applications.

On a much larger scale there are many occasions when part of a structure contains closely spaced and regularly repeated geometrical features; a perforated boiler tube plate being an example that can be visualized. In such cases, it is not sensible to model all the holes individually, as this would require many elements. Nor would it be correct to ignore their presence, as they have a significant effect

on the structural behaviour of the plate. The solution to this problem is to model the plate as a single continuum with equivalent elastic constants that take account of the holes. A finite element modelling procedure can be used to determine these equivalent properties (NAFEMS, 1986). Having obtained a solution, by modelling the plate model as a continuum, the results are used to obtain stresses at the local level by analysing separately a hole and a representative surrounding region of the plate. A similar procedure can also be used to model corrugated plates and bolt flanges that contain a series of bolt holes around the circumference.

Finally, note that even if the finite element mesh and the boundary conditions faithfully reproduce the real physical situation being modelled, there could well be some uncertainty in the property data of the material or materials of the structure, and that this often limits the accuracy of the solution. Hence, the quality of the material property data can have a direct influence on how much effort the analyst needs to spend in producing an accurate finite element model.

6.2 FINDING THE BOUNDARIES OF THE MESH

When calculating the required solution, the governing partial differential equations are solved subject to the appropriate boundary conditions. During the specification of the physical problem the boundaries are defined in terms of the geometry of the structure, but now these boundaries must be found in terms of the mesh that is being used to describe the structure. This involves defining the boundaries as a collection of element faces before, possibly, listing the boundary nodes.

6.2.1 Boundaries of Meshes with a Regular Structure

If the mesh has a regular structure, a knowledge of the local coordinate system (see Sec. 5.10.1) can be used to define a set of indices i, j, k. These indices denote the position of an element within the mesh structure and range from unity to the maximum number of elements in each of the local coordinate directions. The local coordinate system can also be used to define the faces of an element within the mesh. Figure 6.1 shows mesh with a regular structure shown in terms of its local coordinate system. Each element of the mesh has six faces and a typical element is shown with its faces labelled with the points of a compass. Hence the faces are named North (N), South (S), West (W), East (E), Top (T) and Bottom (B). These latter two are also known as High (H) and Low (L) in some programs. For example, the face of the element that is at the most positive local x-direction position, in the direction of increasing the index i, is the East face and the one at the most negative local x-direction position is the West face.

It can also be seen, by looking at Fig. 6.1, that any plane of elements has a constant value of either i, j or k, and that the extent of the plane can be defined by knowing the limits of the other two indices. The patch of elements shown in Fig. 6.2 has a constant value of the index i and the limits are defined by j_{min}, j_{max}, k_{min}

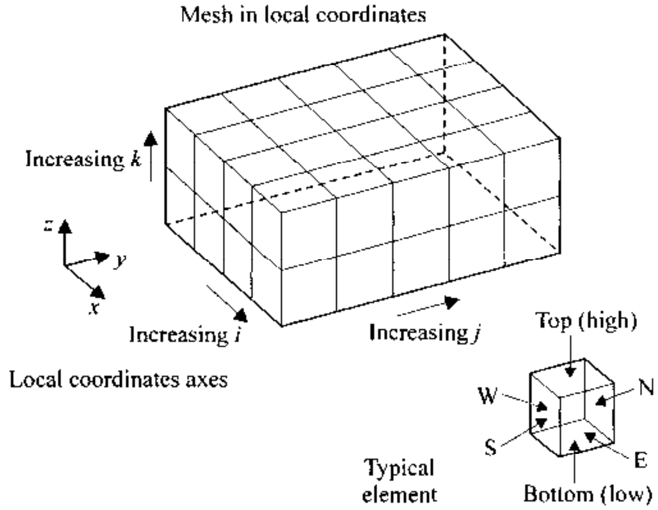


Figure 6.1 Use of a local coordinate system.

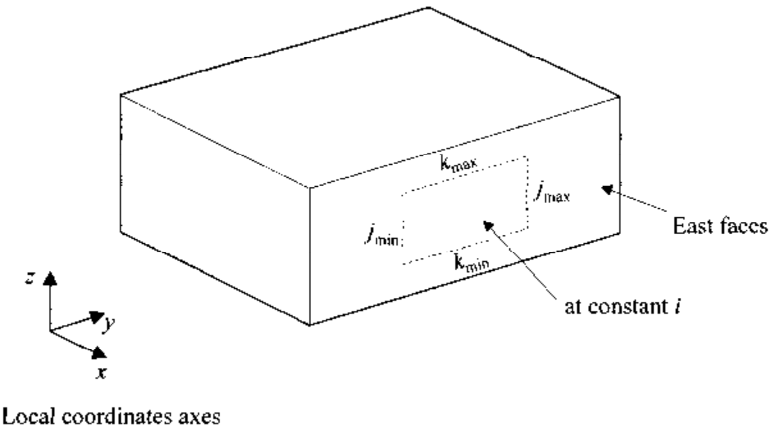


Figure 6.2 Defining a patch of cell faces.

and k_{max}. Also, the faces of the elements in the patch shown are in the positive local x-direction and so they are all East faces. By using this notation a set of patches can be defined on the boundaries of the mesh. These patches have to be defined for all the surfaces where the boundary conditions are to be specified.

Remember that when defining a patch of element faces on a boundary, it is sensible to define patches that have only one boundary condition type applied on the patch. This means that the faces or nodes of the whole patch might be, say, restrained in some direction or have a constant pressure load applied. By doing this it is simple to specify the boundary condition that applies on a patch by a single command.

6.2.2 Boundaries of Meshes with an Irregular Structure

When a mesh has an irregular structure the problem of defining the boundaries becomes much more difficult. Actually finding the element faces that are the boundaries of the mesh is quite straightforward, as will be seen. It is the collecting of the various element faces into groups that are suitable for the addition of the same boundary condition that is difficult.

Two pieces of information help us to find the element faces that are on the boundary of the mesh. First, each face of an element is uniquely defined by the nodes that are on the face and, second, the faces on the boundary of the mesh can be associated with only one element, while those internal to the mesh must be associated with two or more elements. This is shown in Fig. 6.3, where it is clear that the internal face is common to the two elements and that the external faces are related to only one of the two elements.

The process of finding the faces that are on the boundary of a mesh is called a *free-face check*. The algorithm used to do this is shown in Fig. 6.4, from which it can be seen that each element is considered in turn. Then each face within an element is found in terms of the numbers of the nodes attached to it. A unique label for each face on the element is found from these node numbers. Each of these face labels is checked against a list of the face labels stored in a database. This database is created as the process is carried out and records the number of elements to which a given face is attached. If a face label does not exist in the database then an entry recording the new face label is made in the database and the count of occurrences of the face is set to unity. If the face has been listed before, the count is increased so that it reflects the number of elements associated with the particular face. Once all the faces on a element have been processed then a new element is chosen, and after all the elements have been processed the database is complete. By checking the database, a list can be made of all those faces that are attached to only one element. These must be the faces on the boundary of the mesh, and the list of faces is known as a *free-face list*.

Once the free faces have been identified, they can be grouped into the required sets of faces for the different types of boundary conditions. This is usually done by

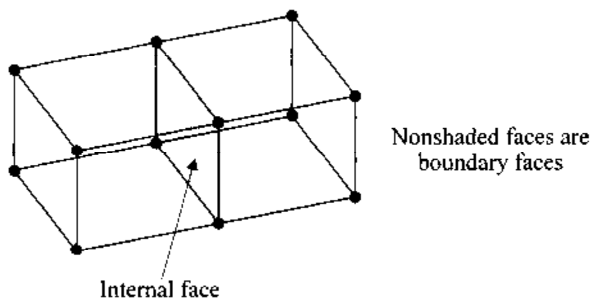


Figure 6.3 Internal and boundary faces.

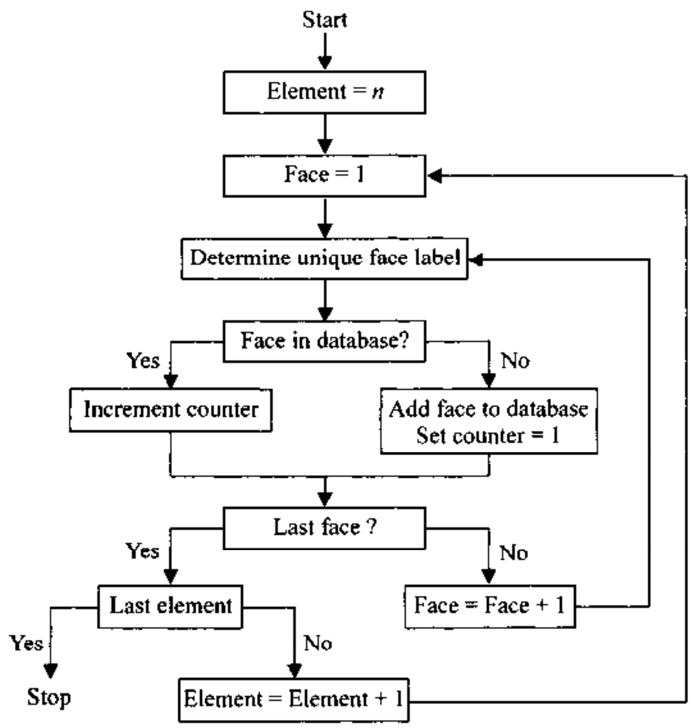


Figure 6.4 An algorithm for the determination of free-faces.

displaying the faces in the free-face list on a graphics screen in a variety of ways, including the following:

- A hidden-line display, where the user sees the faces just as they would be seen if they existed physically. That is, faces that are behind other faces, as seen by the viewer, are hidden from view.
- A display of the faces within a given volume.

Once the displays of the bounding faces of the mesh have been produced the pointing device of the terminal or workstation can be used to pick out the faces. Either this can be done face by face, or whole sets of faces can be picked by placing a window on the screen and noting the faces that are within the window. This is illustrated in Fig. 6.5 which shows a simple mesh made from two mesh blocks of regular structure. Here it is desired that a constant pressure boundary condition is applied at the nine element faces labelled with arrows in the left-hand view. These faces may be picked manually using the cursor on the display screen, but, by changing the view of the mesh to that shown on the right-hand side of the figure, a rectangular window can be defined using two corner points as shown. Then all the faces that are wholly within the window, the nine required faces, can be labelled by the pre-processor as being a set of boundary faces. This windowing

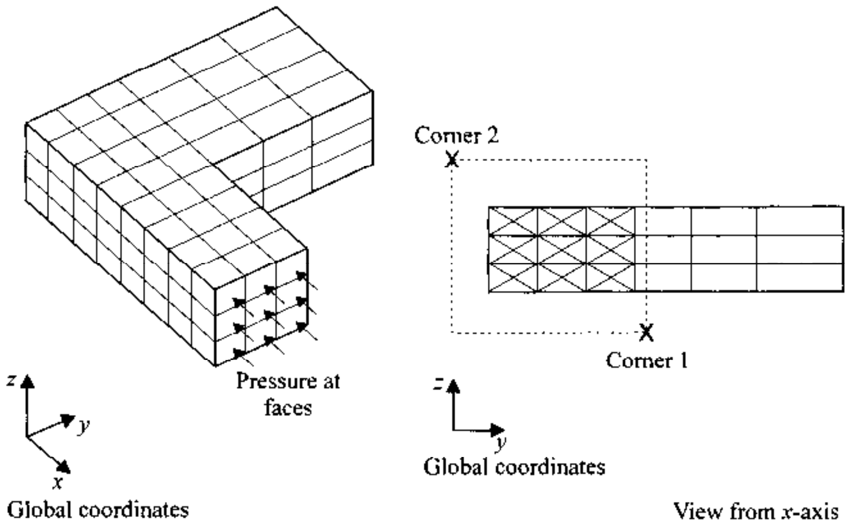


Figure 6.5 Finding boundary faces on the screen.

method has great advantages when dealing with large numbers of boundary faces.

6.2.3 Grouping Faces at the Boundary Together

Regardless of whether the mesh has a regular or an irregular structure, the boundary faces must be grouped together into sets of faces using the methods just described. Each set of faces can then be given an index that allows the set to be related to a boundary condition. Sometimes, the boundary condition on a set of faces is unique to that set, however, in some cases, the same boundary condition may well be applied to several sets of faces. In this latter case, each of the sets can be given the same index and then the index can be linked to the given boundary condition.

6.3 IMPOSING THE BOUNDARY CONDITIONS ON THE MESH

Once the boundaries of the mesh have been identified the appropriate boundary conditions can be applied at these boundaries. For a linear static analysis the displacements of the nodes resulting from a series of forces applied at the nodes are calculated. Hence it is obvious that a series of consistent nodal forces must be specified, which can be done either directly or indirectly as a pressure loading. Less obvious is the requirement that some displacements must also be specified. These must at least restrain the structure from rigid-body motion where the whole structure has a constant load applied to it and accelerates uniformly through space.

One rule that must be kept in mind is that, at the nodes where restraints are applied, the associated nodal forces cannot also be applied as the numerical solution will overwrite the values of the force at these nodes to ensure that the correct displacement is calculated.

6.3.1 Point Forces

In the real world forces are applied to a structure through a contact patch on the structure. Even when the loads are effectively point loads this patch must have some finite area and so the modelling in the finite element analysis is an approximation to reality. Hence the analyst must consider in detail how to represent the physical problem through the finite element model.

To represent even point forces, therefore, the appropriate load might have to be applied over several nodes depending on the size of the elements in the region of the contact. For example, Fig. 6.6(a) shows the physical loading of a two-dimensional structure, together with two ways in which the load could be modelled. In Fig. 6.6(b), the mesh is sufficiently fine in the region where the

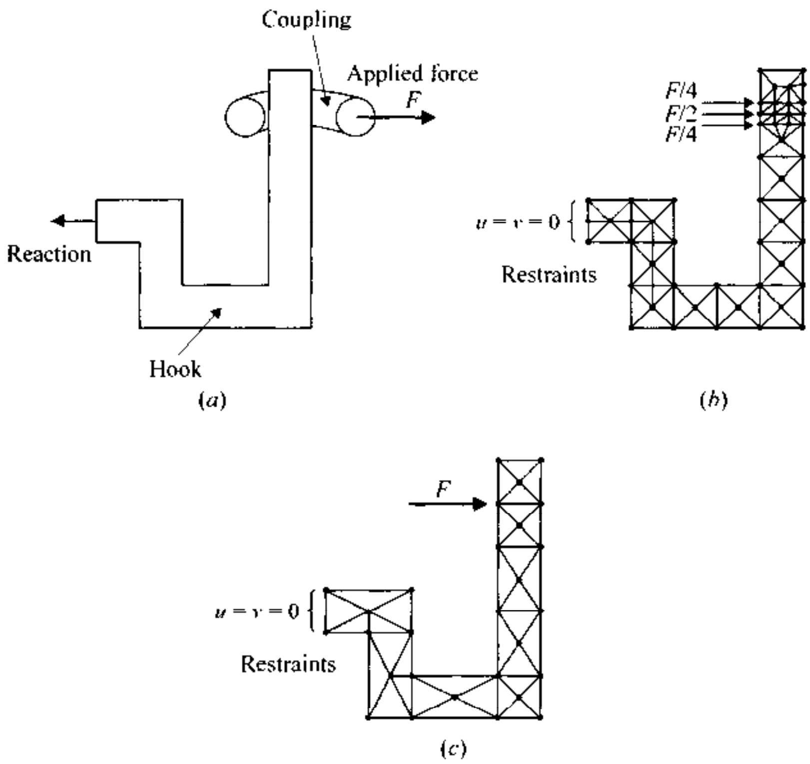


Figure 6.6 Forces on a hook: (a) the situation, (b) with a fine mesh, (c) with a coarse mesh.

force is applied and so the load has been apportioned over three nodes, whereas in Fig. 6.6(c) the mesh is coarse and the load has been applied at just one node. In a similar way distributed loads can also be applied over a surface. For example, a constant pressure load may be represented by an averaged force at a series of nodes.

Clearly, for both two- and three-dimensional problems, the forces will be defined using appropriate components in, say, the x-, y- and z-directions.

6.3.2 Pressure Loads

On element faces or edges on a boundary pressure loads can be applied by most programs. These require just the pressure on the element face or edge to be specified. From the element geometry, the face area or edge line for a plate or shell element can be found and hence appropriate nodal loads can be applied on the element face or edge, or on a series of linked element faces or edges.

It is known that modelling errors are introduced, when applying pressure loads, if the real geometry differs from that of the mesh and the elements are distorted from their parent shape.

6.3.3 Fixed Displacements

To prevent the structure from moving freely in space, which is a numerically singular problem, it must be restrained in all the coordinate directions at one node at least. This means that a displacement of zero must be specified in all the coordinate directions for one node. Sometimes whole sets of nodes need to be restrained in all directions to simulate a rigid fixing such as a floor mounting. An example of this is the restraint applied to the left-hand end of the hook shown in Fig. 6.6. There the nodes are restrained by setting the displacements in both coordinate directions to zero. Implicitly, the displacement in the third coordinate direction is set to zero for a two-dimensional program, but when a three-dimensional program is used the analyst must remember to set the displacement to zero explicitly.

A common use of fixed displacements is in the simulation of sliding, where nodes are free to move in one direction but are restrained to have zero movement in all other directions. This is shown in Fig. 6.7 where sliding of the arm in the vertical direction is allowed. Here the z-direction is the vertical direction and so the displacement of the upper end of the arm in both x- and y- directions is set to zero. Equally all three displacements at the pinjoint are set to zero.

6.3.4 Fixed Rotations

As well as fixing displacements, the rotation at some nodes may also have to be specified. Examples of this will be given in the examples in Chapters 7–9.

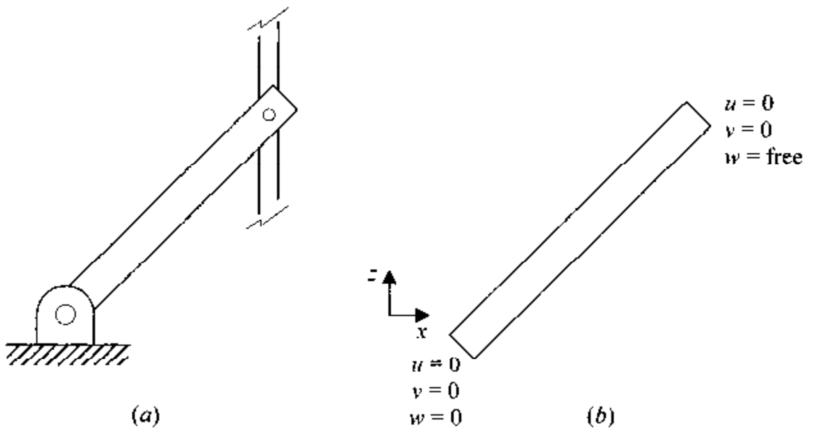


Figure 6.7 Example of applying restraints: (*a*) the situation, (*b*) the restraints applied to model the situation.

6.4 CONTROLLING THE SOLUTION

At this stage in the analysis process the model data consists of the finite element mesh defined by the nodal coordinates and the element connectivity, the material properties for the elements and the boundary conditions in terms of forces and displacements at appropriate nodes. Whether this data is stored in the database of a pre-processor or as a collection of records in a neutral (ASCII) file it must now be made available for the solver to read. For a linear static analysis the only other parameters that the solver might need are the method of solving the linear equations and any associated parameters. For example the solver may have to be instructed to use a direct solver or iterative solver of a particular type. In the former case no further information is required but in the latter case the number of iterations and any relaxation parameters need to be specified.

For more complex analyses such as those discussed in Chapter 10 other parameters such as the initial conditions, the number of iterations to remove nonlinearity, stress or displacement constraints for optimization problems and the method of eigenvalue analysis are required. Finally, in all cases, file management data is needed to specify the source of the input data for the solver and the destination of the output data from the solver. One useful file handling device that can be employed is to store the upper and lower matrices, decomposed from the global stiffness matrix, on disk. This enables further simulations using the same mesh, with say different loading conditions, to be run without the computational expense of recalculating these matrices.

6.5 RUNNING THE SOLVER

Once all of the model data and management data have been assembled, it is necessary to issue the command that runs the solver program. This can be done manually, once all the input files are in the correct place in the directory structure of the correct computer, but many applications now make use of a simple command file written in the operating system language of the machine where the pre- and post-processing takes place. This command file performs the following functions:

- It copies the input data for the solver to the correct directory (and correct computer if necessary).
- It tells the solver where the find the input data.
- It runs the solver program.
- It tells the solver where to write its output data.
- It copies the output data, both results and management data, to the correct directory (and correct computer if necessary).

Using such files the process of running the solver becomes extremely simple for the analyst.

6.6 TROUBLESHOOTING

Once the solver has run the results should be available for the analyst to check. However, it is very common for analysts to check for results and find none available, perhaps for the following reasons:

- The solver has not finished its calculations.
- The solver has encountered an error when reading the input data, such as the input data files not being in the correct location or the data files being in an incorrect format.
- The solver finds that no solution can be found to the linear simultaneous equations owing to incorrect data such as a mesh with coincident nodes or collapsed elements or inadequate restraints on the boundary.
- The solver encounters an error when writing the output data, for example when the target disk for the output data is full or the target directory does not exist.

Consequently an analyst should do the following:

- Check to see that the solver has finished by interrogating the computer's list of processes running at any one time.
- Look at any error and warning files written by the solver to see if there has been a failure. It is here that both file management errors such as incorrect directory specification or disk full errors as well as errors in solution will be found. Also the solver may write warning messages to alert the user to potential inaccuracies.

- Check and, if necessary, correct the command file.
- Check the integrity of the mesh in terms of coincident nodes and spurious free faces and modify where necessary.
- Check the material and physical properties of the elements, and modify where necessary.
- Check the boundary conditions to ensure that the correct forces in the correct directions are applied and that the structure is adequately restrained to remove numerical singularities due to rigid-body motion.

When all these have been done, and any errors corrected, it should be possible to resubmit the command file and find that on completion the solver has produced the necessary results files. Often this does not happen and so the checking process might need to be repeated and further corrections made. Even when results are produced they may not be of any use and determining whether this is the case or not is the subject of the next few sections.

6.7 THE RESULTS GENERATED BY THE SOLVER

When the solver runs it produces a large amount of data that has to be analysed. This analysis is undertaken so that the quality of the solution can be examined and so that useful technical information can be extracted where appropriate. First, it must be decided what information will actually be available at the beginning of the results analysis.

Information can be produced by the solver in two main forms. These forms differ in how the data is stored by the computer. In one form the data is stored using an internationally agreed format that defines individual characters of data such as the letters of the alphabet or the numbers 0 to 9. This form of data is known as *ASCII data*, after the committee that divised the data standard, and can be written to a terminal screen or stored in a file known as an *ASCII file*. Each character has to be defined by one byte, i.e. 8 bits, of computer memory and so 256 different characters can be specified. ASCII files of data can be edited by text processors and other software, and they are effectively machine independent which means that the data can be transferred from one computer to another computer, even if the machines are from different manufacturers, without any translation process taking place. While most computer manufacturers use the ASCII standard, there are other standards such as EBCDIC which are used by a minority of manufacturers.

Numerical data can also be stored in the second data storage format, which is known as *binary data format*. There is a standard for this method of data storage, but usually, at present at least, the method of storage is peculiar to each computer operating system or computer manufacturer. Each of these binary storage methods enables real numbers, for example, to be stored by four bytes in single precision or eight bytes in double precision. Binary data is stored in files known as *binary files*. These files are not machine independent and so cannot be transferred

from computer to computer without some form of translation process taking place. Sometimes when a workstation, for example, is connected to a minisuper-computer a translation program is provided by the workstation vendor to facilitate the transfer process. By using binary files to store real numbers, there is a saving in the amount of storage required, as can be seen from the number of bytes required to store each number.

The type of information produced by the solver program can usually be controlled by the user but it often consists of the following:

- Values of the residual error if the linear equations have been solved iteratively. This gives a measure of the accuracy of the solution in numerical terms on the mesh used.
- A complete list of the variables at all the nodes of the mesh or all the elements of the mesh. Typically the list consists of nodal displacements, and stresses and strains at the nodes.
- Mesh data. This is usually produced by the pre-processor but may be modified by the solver program if some form of automatic mesh adaptivity has been carried out where extra nodes and elements have been added to the mesh or where nodes have been moved to try and improve the accuracy of the solution. It will include the coordinates of the nodes and the element connectivity list.
- Some form of ASCII file that reports on the progress of the solution. This file might include an echo of the input data from the pre-processor so that the input actually used by the solver can be checked, a repeat of the residual values, if there are any, and any user-programmed results, such as the displacements or stresses of a given node or the reaction forces. Accounting information such as the length of time that the solver took to run and the amount of disk resources used may also be listed.

Here, the concern is with how the actual structural data at all the nodes and elements can be analysed. Large quantities of this data are produced by the solver, especially if the mesh is complex and has a large number of nodes or elements, as is generally the case for an industrial problem. Only when small test cases are run is it possible to read the ASCII files that contain the solution and so for realistic problems computer graphics techniques are used to analyse the results visually.

6.8 USING COMPUTER GRAPHICS TO EVALUATE SOLUTIONS

6.8.1 Using Graphics Hardware

Before considering what can be done with computer graphics let us think about the hardware that is required to drive the software that generates the pictures as well as to display the pictures themselves. A typical hardware installation consists of the following devices:

- A screen or visual display unit (VDU) that is able to produce a grid of points in a variety of colours. These points are known as pixels (see Sec. 4.3.2). The resolution of the screen is determined by the number of pixels that can be displayed; most graphics screens can display a grid of 1000 pixels in the horizontal direction by 1000 pixels in the vertical direction. If the display is monochrome then each pixel can be shown as either black or white, whereas if the display is a colour device then each pixel can be displayed in one of several colours. Typically, 16 colours or even 256 colours are used. The screen could be part of a terminal attached to a computer or it could be part of a workstation.
- A keyboard that allows the user to interact with the software by typing commands and replying to questions from the software.
- A pointing device that enables a cursor to be moved around the screen. This pointing device may be a *mouse* which is a small device that senses movement either mechanically or optically, or it may be a simple set of four direction keys.
- A button box. This is used in the more expensive installations to manipulate the picture. The box has several knobs on it that can be used to rotate an existing picture about any of the three coordinate axes, or to zoom in and out or pan across the picture.

When the user runs the graphics software, the program activates the screen, keyboard, pointing device and button box in such a way that the user can develop an intuitive feel for the manipulation of the results.

6.8.2 Using Graphics Software

The graphics software itself is usually supplied as part of the structural analysis software package and is known as a post-processor. Sometimes, however, this software is combined together with the pre-processor to form a single interactive program that is used for both creating the computer model and post-processing. These programs enable a user to see the geometry of the structure, the mesh and the results of the simulation by producing pictures of the available data, usually in colour. Displaying the data in a visual way condenses the vast amount of information that a solver can generate into a usable format. As computer power becomes cheaper, graphics software is often run on interactive colour workstations that have sufficient display resolution for the task and also have enough of their own computer power to produce detailed pictures in a reasonable time without having an impact on other users on the network.

By entering commands the analyst can drive the post-processing software. These commands direct the software to build the required picture of the data on the graphics screen. Several commands may be needed to create a picture and, in many cases, the analyst will want to generate similar pictures from one analysis to the next. To prevent the user from re-entering a lengthy set of commands it is often possible for the software to read the commands from an ASCII file. This file

can be created by the user with a text editor or it can be written by the software itself in some cases.

When generating the pictures, the stages that are followed are similar regardless of the type of data being displayed. The display process involves, first of all, displaying some part of the geometry or mesh on the screen. This may be a collection of the basic entities that comprise the geometrical hierarchy (see Fig. 5.5), or the boundaries of the mesh or even some part of the mesh itself, in either its original or its deformed state. Then, the picture is manipulated so that the required view is displayed before the solution itself is shown. This final display might be some of the stress data, shown as a set of vectors, or the contours of scalar variables such as the von Mises' stresses or even a magnified view of the displaced geometry. These three stages—show the geometry, modify the view and display the results—can be performed in any order but it is usual to display the actual results last of all. As this post-processing part of the analysis process is highly interactive, the user can often move between these three stages in a seemingly random fashion. However, for most simple cases, it is most useful if the order given above is followed. The following sections deal with each of these three stages in turn.

6.8.3 Plotting the Geometry

When the post-processing software is started it has to read the files of results and mesh data. Then the user has to find the required view. One way of doing this is to plot some part of the geometry, normally a part of the mesh used in the solution process, onto the graphics screen. This can be done by asking the program to display the basic entities used to create the mesh, if it has access to this data, or the boundaries of the mesh or the mesh itself. Exactly which of these is used depends on the capabilities of the software itself and the user's preference. A simple plot of the boundaries of the mesh is usually good enough at this stage.

However, a plot of the geometry or mesh can be used to check that the geometry looks like the physical situation and also to check the integrity of the mesh (see Sec. 5.13). 'Integrity' means that the mesh should both represent the required domain and be structured in the correct way. The display of the mesh shows a user the elements that have been used in the calculation procedure, and so any significant errors in the mesh or bad modelling practice can be found.

The way in which the mesh is displayed depends on the mesh structure that is being used. If the mesh has a regular structure then the local coordinate system and the node or element indices can be used to specify areas of the mesh just as was done in Sec. 6.2. Sheets of element faces can be defined in this way and then displayed. On the other hand, if an unstructured mesh is being used then the elements can be grouped in some way and the group projected onto some cutting plane in space. Another way of displaying the mesh is to draw only the free faces of the mesh.

6.8.4 Obtaining the Required View

Once the geometry has been plotted, the view of the geometry may well have to be manipulated. There are an infinite number of ways of looking at any image and so there must be some means of defining the exact view that is required. The picture on the screen is drawn as if a single eye is looking at the object being drawn. This situation leads the graphics software to require the user to define a few fundamental pieces of data. This data can include such things as:

- the *target point*, which is the point in space at which the eye is looking
- the *eye position*, which is the point in space at which the viewing eye is placed
- the *up-direction*, which defines where the top of the picture should be
- the *viewing area*, which enables the apparent size of the objects in a view to be specified.

In Fig. 6.8, the target point is taken to be at the origin of a set of Cartesian axes. This target is shown being viewed by a single eye which can be placed in two different positions. Default values are always given by the software for the initial specification of both the target point and the eye position. These could be the origin and a point on the x-axis, such as eye position 1, respectively. When plotting data that relates to engineering work, the eye is normally at an infinite distance from the target and so the effects of perspective are not seen. This means that even though the eye position can be defined as a point in space the software actually places the eye at infinity on the same directional vector that joins the eye position and the target position. So, it can be seen that it is the combination of the eye position and the target point that defines the vector along which the eye looks. For some work, however, such as architectural drawing or aesthetic design, perspective effects can be produced by the software and then the eye position is the actual point in space at which the eye is placed.

Defining these two positions in space is still not sufficient to specify the view of an object. Human beings have a sophisticated balance system and this gives us

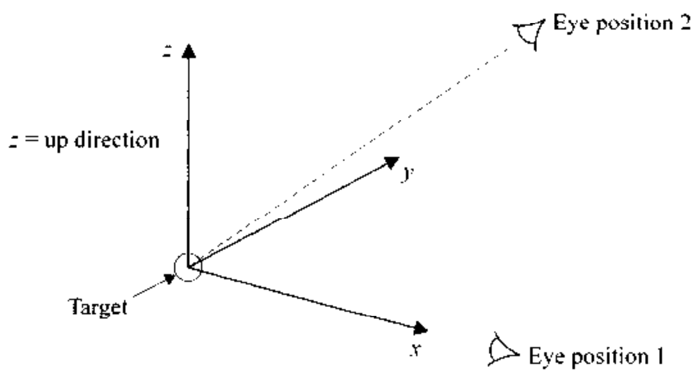


Figure 6.8 Target and eye positions.

information as to which is the vertical direction and so where up and down are. Computers are not as sophisticated and so they have to be told where the vertical direction is. This direction is also known as the up direction. In Fig. 6.8, the up direction is in the positive z-direction. Consequently, the post-processor must be told which direction the up direction is, if the pictures that it produces are to have the structure in a realistic orientation. One command usually enables this direction to be specified.

If the up direction cannot be specified to the post-processor, as is sometimes the case, then the picture has to be orientated by a series of rotations about the three coordinate axes. This is usually achieved by specifying the angles for each global coordinate axis, x, y and z, through which the axes are to be rotated. It is difficult to produce the correct view this way using a single command. Several attempts may be needed to get the picture right.

Once the eye position, target point and the picture orientation are known, the display software can take the three-dimensional data for the geometry or mesh and draw it on the screen, in what is, of course, a two-dimensional representation. This can be done in one of two ways. The original way was to transform the three-dimensional data into two-dimensional data using the post-processing software. This two-dimensional data can then be plotted. Many systems still use this technique, but a more recent way of handling the data is for the post-processing software to send the three-dimensional data to the display hardware, together with the current eye-position, target point and the vertical orientation. The transformation of the data from this set of three-dimensional vectors into a two-dimensional picture is then carried out within the hardware itself by a combination of both hardware and software, known as *firmware*. This local transformation is extremely fast as the firmware is dedicated to the task. Once the three-dimensional data is stored by the firmware it can be manipulated into further displays very easily and quickly, and this is where the button box, mentioned in Sec. 6.8.1, can be used very effectively to modify the target point, eye-position or orientation, signalling the firmware to produce the new pictures so fast that the objects can be moved in real time.

Quite often, attention must be focused on one particular area of the model, for example to see the detailed maximum principal stress pattern at a corner of an object. This can be done by changing the target position and the viewing area. The mechanics of doing this with the post-processor can vary, but there is nearly always a *zoom* command or a *centre* command. Figure 6.9 shows an example of the zoom command being used. This allows a rectangular window to be placed over the current view by defining the two ends of one of the diagonals of the window with the cursor. The software then modifies the target position and the view area to display the picture within the limits of the window. This is done while ensuring that the aspect ratio of the geometry is preserved. The centre command works in a similar way, but the user has to define the required centre of the new view, together with the magnification required, as shown in Fig. 6.10. By using these commands in the correct combination, the view of a mesh or the results can be infinitely varied. There are also ways of returning to the original view.

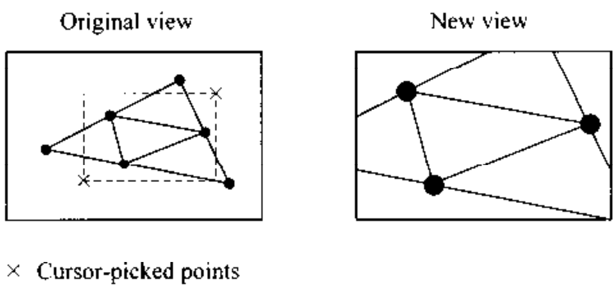


Figure 6.9 Changing the view by a zoom operation.

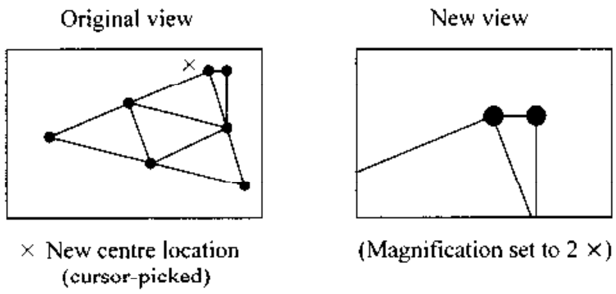


Figure 6.10 Changing the view by moving the centre of the display and applying a magnification factor.

When working with very complex meshes, and the associated results, the sheer volume of information displayed can be too great. The information content can be restricted by using the following techniques:

- *Volume clipping* This enables the user to give limits in the global coordinates x, y and z within which objects are displayed, but outside of which they are ignored.
- *Suppression of hidden lines* This calculates whether something that would be drawn is hidden from view by any other object, such as an element face. If the object is hidden from view it is not drawn. The displays that are generated using this method are often called hidden-line displays.

6.8.5 Displaying the Results

Having looked at how the geometry or mesh of the model can be displayed, and knowing how to orientate the view to give the desired picture, some of the results can be added to the picture. The results that can be viewed graphically have to be derived from the nodal displacement data.

With a mesh that has a regular structure the results data can be drawn for a sheet of elements or nodes, in the same way as the mesh can be drawn. Remember

that this sheet may not be planar in global coordinate space, as even a mesh with a regular structure can be curved in space so that it fits around an object. When the mesh has an irregular structure the display of results is not so straightforward. As there is no simple way of referring to a group of elements, many post-processors allow the user to define a geometrical plane through the mesh onto which the results are interpolated. This plane is known as a *cutting plane*. Other ways of grouping elements can also be used, such as showing a hidden-line plot of the results which displays only those results on the boundaries of the mesh, or displaying the results for a restricted group of elements defined by creating a list of element numbers.

No matter which way is used to display the data, there are essentially three types of results display:

- *Vector plots* show the vectors relating to the stress results.
- *Contour plots* show contours of the scalar variables over the domain.
- *Deformed geometry plots* show the deformed geometry in a form that magnifies the displacements.

The last case is simple, but the first two need some explanation. Note that the last case can be combined with either of the first two cases.

Dealing with vector plots first, the vectors are displayed within the picture as arrows in two dimensions. These plots are what are seen when the so-called wind arrows are shown on weather forecasts. Plotting stress information in this way can lead to confusing displays being produced as information is lost. The arrows that are drawn are the projections of a three-dimensional vector into two dimensions. Take, for example, a vector pointing directly out of the page; this would be displayed as a point. So that some of the lost information can be retrieved, the arrows are often colour coded to denote the absolute magnitude of the vector. Usually, red denotes a high stress and blue a low stress with intermediate shades denoting the stress values in between.

One other problem that has to be dealt with concerns the length of a typical vector arrow. Depending on the problem, the user will want the length of the arrows to give as informative a display as possible. This means that the user must scale the arrows appropriately, either by letting the computer draw some arrows and then scaling them, or by giving the computer a typical stress which might represent, say, 10 per cent of the screen width.

For meshes that have very dense element distributions the arrows may be so close together that too much is displayed and the useful information is obliterated. This can be overcome by the software interpolating the stress data onto a coarse, regular grid of points. The user specifies the distance between the points in the grid, and the arrows are drawn at the points. One problem with this type of display is that the true nature of the computed stress field can be hidden from the user. Sometimes it is better to display the data at the positions at which it was calculated. (In later chapters vector plots have not been used for the reasons above as well as the fact that monochrome vector plots are extremely difficult to interpret.)

Contour plots are pictures of the lines of constant scalar value of some variable plotted through the domain. They are similar to the isobars that are seen on maps for weather forecasts. Little interaction is required to produce these plots, except perhaps to specify the number of contours that are to be drawn and their values, which need not be specified with a constant increment. Typically, about 10 contours are calculated, and again these are colour coded in the picture to show the value of the variable on the contour. A coding scheme similar to that used for the magnitude of a vector is used in this case as well. Sometimes, the contour levels can be chosen by the user to give the required values. This is done where several separate pictures of contours have to be produced to create the required display, and it provides a consistent display.

A variation of the contour plot is to use a surface plot. This is generated by displaying a three-dimensional surface, the height of which above a plane is a measure of some variable. This variable should be a function of the two dimensions that describe the plane. In effect, the display shows a series of mountains and valleys.

6.9 CHECKING A SOLUTION FOR ACCURACY

When analysing the results of a simulation, certain pieces of information are required. For example, a prediction of the von Mises' equivalent stress at some point in the structure may be needed.

As a check on the quality of the result:

- The stress field and displacements should look qualitatively correct. If these simple checks show that there might be problems with the quality of the results then users should consider checking their input data and changing their models, if necessary, before re-running the solver program.

6.10 MAKING REFINEMENTS

If it looks likely that a model must be refined, a user must consider the advantages of producing a better prediction against the cost constraint of repeating the whole simulation process. Quite often even crude models can give large amounts of new and useful information to a user. This might prove adequate for the purposes of some users but not for others. It all depends on the application under consideration.

The process of refining a model can include increasing the density of nodes in a given area so that the changes of the displacement and hence stress in that area can be more accurately captured, for example, in the region of a hole or other structural discontinuity. In terms of effort, this involves a large amount of work, as it involves rebuilding the mesh of the domain, either by repeating one of the mesh generation processes described in Chapter 5, or by using an adaptive

meshing process. Once the mesh is built the problem specification within the pre-processor and the setting of the boundary conditions has to be carried out again, transforming the data generated as part of the original specification process onto the new mesh.

A systematic way of increasing the mesh density for a mesh with a regular structure is to double the number of elements in each of the local mesh directions. Similar refinement schemes can also be carried out with unstructured meshes, for example, by placing a new node at the centroid of each element and then remeshing. With the new mesh a solution is calculated, and the results obtained. When the results do not vary in global terms from one mesh refinement to the next then the results are said to be mesh independent. While it is always desirable for the results to be independent of the mesh size, for many industrial problems this is not always possible as the constraints in terms of cost or time or computer capacity are too great.

6.11 FAILURE CRITERIA

Stress results from an analysis can be used to determine whether or not the material has failed. This means that, for a successful design, the geometry of the structure has to be modified to relieve damaging stress concentrations. This failure analysis is usually achieved by transforming the nodal stresses, given with respect to the global coordinate directions, into the principal stresses and substituting these stress values into a failure criterion appropriate to the material in the structure.

The most common of the failure criteria found in packages is that due to von Mises. It is a criterion relevant to ductile materials, such as metal alloys, that fail by the single mechanism of a material yielding. Output from the criterion is in the form of a stress, known as the von Mises' stress, which, if it exceeds an allowable material stress (often the tensile yield strength), tells the analyst that the material has failed.

For further details on the formulation and application of failure criteria for structural materials consult texts specific to the material in question. Often the software manuals provide an introduction to the failure criteria available to the analyst, but be aware that this information does not generally go into sufficient depth to mention any pitfalls that might occur when these criteria are used such that the limitations of the criteria are ignored.

7

SOME EXAMPLE PROBLEMS

This chapter considers example problems that illustrate the techniques discussed in the previous chapters with regard to small displacement linear elastic analysis. Simple examples will be discussed before moving to the more complex examples of Chapter 8 and the industrial examples of Chapter 9. The effects of dead weight and real imperfections have not been included in the examples.

7.1 THE BRAZILIAN OR SPLIT TENSILE TEST

This problem is contrived to be two-dimensional and of plane stress type. It illustrates a simple use of the finite element method with an isotropic homogeneous material. Further, the equations developed in Chapter 2 yield a classical elasticity solution against which the finite element solution can be compared. There is also a physical, practical use to this test as it is a standard test method for the tensile strength of brittle materials.

Through this example it is hoped to demonstrate the use of symmetry, the application of a concentrated point load which is a stress singularity, the mesh specification for an unstructured, or free, mesh using triangular elements, the modelling of curved boundary by quadratic elements, the grading of a mesh for variable stress gradients and convergence by p-refinement methods.

Over the years a considerable amount of experience has been gained by the authors by using this example as the subject of undergraduate assignments. It is particularly suitable for this as any finite element solution can be compared with both the classical solution and the behaviour of specimens in physical tests. From this comparison the success of finite elements in modelling the problem is easily demonstrated.

7.1.1 Problem Definition

Figure 7.1 shows both the geometry and the loading boundary conditions for the Brazilian test. A disc of material is compressed by diametrically opposite concentrated (or point) loads, where d is the diameter of the disc and P is the vertical load acting along the line AOB. Here, it is assumed that the diameter d is very much greater than the thickness of the disc t such that the problem can be reduced to a two-dimensional case with plane stress. Furthermore, it is assumed that the values of P, d and t are such that there is no nonlinearity induced in the material and so the problem can be modelled as being one with linear elasticity and small displacements. Hence a linear static analysis can be performed.

Although the presence of circular symmetry suggests that the use of a polar coordinate system may be convenient, the results presented here are defined with reference to a Cartesian coordinate system placed at the origin O. Note that the planes of mirror symmetry for this loading case are labelled AOB and COD.

In practice, the Brazilian test method is used to determine the tensile strength of brittle materials such as ceramics and concretes, because a favourable stress distribution is generated in the specimen. In such a test, a disc (or cylinder) is compressed between two hard parallel metal platens, using a standard testing machine, until ultimate failure occurs and the specimen splits along diameter AOB.

This example is suitable as an undergraduate assignment where plots (in a suitable non-dimensional form) of the stresses σ_x, σ_y, τ_{xy} along the diameters AOB and COD have to be created using both a finite element analysis and a classical elasticity solution.

7.1.2 Classical Elasticity Solution

By following the Airy stress function approach outlined in Sec. 2.3.2, a classical elasticity solution for this problem can be developed provided that the material is

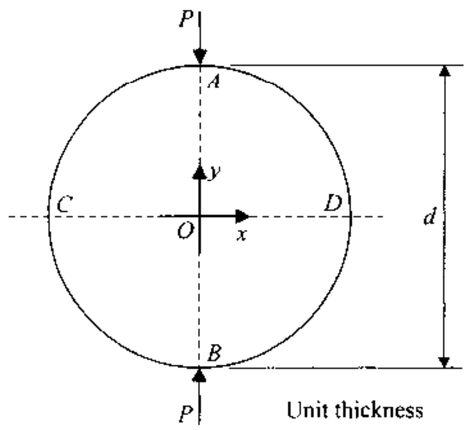


Figure 7.1 The arrangement for the Brazilian tensile test.

homogeneous and isotropic (Timoshenko and Goodier, 1988). It is shown that the stresses (σ_x, σ_y, τ_{xy}) are as follows:

Plane *COD*

$$
\begin{aligned}
\sigma_{y(CD)} \Big/ \left(\frac{2P}{\pi d}\right) &= 1 - \frac{4d^4}{(d^2 + 4x^2)^2} \\
\sigma_{x(CD)} \Big/ \left(\frac{2P}{\pi d}\right) &= 1 - \frac{16x^2 d^2}{(d^2 + 4x^2)^2} \\
\tau_{xy(CD)} \Big/ \left(\frac{2P}{\pi d}\right) &= 0
\end{aligned}
\tag{7.1}
$$

Plane *AOB*

$$
\begin{aligned}
\sigma_{y(AB)} \Big/ \left(\frac{2P}{\pi d}\right) &= 1 - \frac{d}{d/2 + y} - \frac{d}{d/2 - y} \\
\sigma_{x(AB)} \Big/ \left(\frac{2P}{\pi d}\right) &= 1 \\
\tau_{xy(AB)} \Big/ \left(\frac{2P}{\pi d}\right) &= 0
\end{aligned}
\tag{7.2}
$$

Note that $\sigma_{x(AB)}$ is a *constant tensile stress* , and that it is this feature of the stress distribution that makes the Brazilian test so useful in measuring the tensile strength of brittle materials. Also note that the thickness t is taken to be unity and so does not appear in the expressions. Finally, it is convenient to use the nondimensionalized forms given in (7.1) and (7.2) when plotting stress results as these forms are independent of the values of load P and diameter d.

From (7.1) and (7.2) several interesting features of the solution can be seen:

- σ_x is tensile and constant along diameter *AOB*.
- τ_{xy} is zero along diameters *AOB* and *COD*.
- Both σ_x and σ_y vary gradually along *COD*. These stresses are also seen to be zero at points *C* and *D* which are located at stress-free surfaces.
- The value of σ_y tends to minus infinity as the point of loading is approached. This represents a mathematical singularity resulting from the assumption that all the load is applied at a point. Physically this is unrealistic as the load cannot be applied over an infinitesimally small area. This analysis does not account for any yielding of the material, such as would be included to ensure that the area of the loading was acceptable if the material were a ductile metal. In reality, where the infinite stress occurs, local deformation or failure increases the area through which the load P is transmitted, as shown in Fig. 7.2. This results in the realistic situation of a distributed pressure loading being applied over a small finite area.
- The solutions along diameters *AOB* and *COD* are symmetric about point *O*.

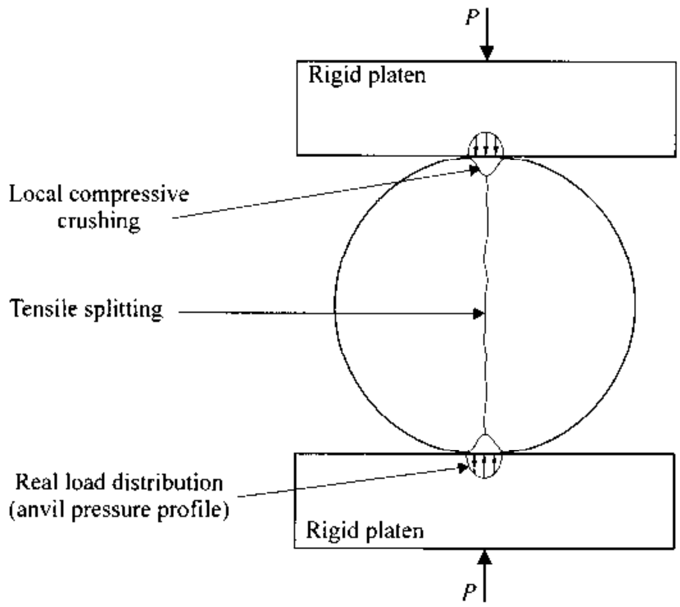


Figure 7.2 Failure of the disc.

7.1.3 Definition of the Finite Element Model

Having looked at some of the background to the Brazilian test and having discussed the classical solution to this problem, the finite element modelling process described in overview in Sec. 4.1 can be begun. The first stage in this process is the initial thinking stage where the structural problem is described (see Sec. 2.4) in terms of the analysis requirements, the structural geometry, the material properties and the loads and restraints on the structure.

As a classical solution to this problem in the form of (7.1) and (7.2) for the stresses along two diameters exists, it is appropriate to build a finite element model and compare the stresses generated by this model with those of the classical solution. More complicated requirements for other problems will be described later.

Turning to the geometry of the problem, Fig. 7.1 shows the actual Brazilian test, but as the shear stresses are zero along diameters AOB and COD, engineering judgement can be used to determine that there is quarter-plane symmetry about these diameters. Hence, only one-quarter of the geometry needs to be considered. For convenience the positive x–y quadrant is chosen as shown in Fig. 7.3. Finally, as a plane stress problem is required, the thickness of the disc in the third dimension is taken to be unity.

This test is used on a wide variety of materials and, looking at (7.1) and (7.2), it can be seen that the stresses should be and are, for the reasons discussed in

Chapter 2, independent of the value of the material elastic constants. This means that it does not matter which material properties are chosen so long as they are isotropic.

As to the physical loads and restraints on the geometry, Fig. 7.3 shows the situation for the quarter disc. If the full load to the disc is P then the model requires a load of $P/2$ to be applied. On loading the disc, the point at the origin O can have neither u- nor v-displacement, whereas any other point along the vertical line OA is free to move in the y-direction but not in the x-direction where symmetry forces the disc to be restrained. Similarly, along line OD any point except the origin is free to move in the x-direction but not in the y-direction. Hence along OA the v-displacement is free and the u-displacement is zero and, along OD the u-displacement is free and the v-displacement is zero.

Having thought about the physical problem, the computer modelling can begin. From Sec. 4.1 the second stage of the analysis process is generating a mesh, although as was seen in Chapter 5 this includes developing a model of the geometry of the structure. In this case the geometry is very simple and can be modelled by two straight lines and a circular arc. Usually, defining the endpoints of the lines and three points associated with the arc, say the centre, start and endpoints is all that is required. To produce the mesh the volume or, in this case, area of the structure must be defined. This is done by forming a closed loop from the two lines and the arc in sequence. For this problem the origin of the Cartesian coordinate system has been taken as point O and the radius of the disc as 500 mm.

Once the geometry has been defined a mesh must be created, but this mesh must be suitable for the problem being modelled. The following are pointers as to what constitutes a suitable mesh:

- The stress gradients (and strain gradients) are going to be very high in the region close to the point of loading and so the smallest elements in the mesh are

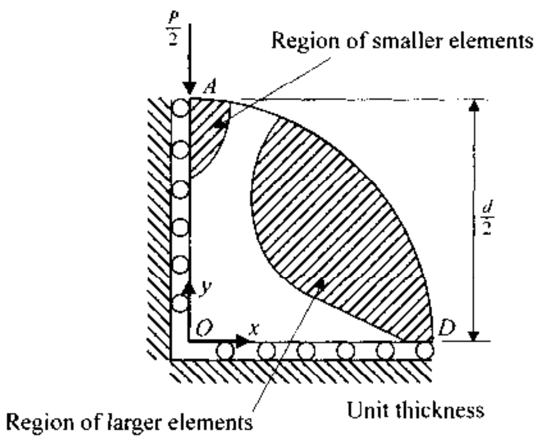


Figure 7.3 Boundary conditions for the Brazilian test.

going to be in this region. Elsewhere the gradients are relatively low and so the element size can be larger. Further, as nodal stress values along the boundaries of the model are to be compared during the results analysis there must be an adequate number of nodes (or elements) along the radii OA and OD. Finally, the curved boundary, except at the load point, is stress free, having zero traction. This means that the largest elements can be placed along this boundary. All of this means that a graded mesh must be used with the smallest elements near the point of loading and the largest elements along the curved boundary, as shown in Fig. 7.3.

- The shape of the geometry, which is topologically triangular, suggests that it might be sensible to use triangular elements. Also, as one of the sides is curved these elements should be of at least quadratic order.

These points indicate that an unstructured mesh is the most suitable, with some form a grading applied to the element sizes. In Sec. 5.9.1 it was stated that, to prevent numerical errors creeping into the solution, it is advisable not to let the volume of the elements in a mesh differ by more than 30 times. To enforce this the typical element dimensions have been specified to be $\frac{1}{20}$ of the disc radius near the load point A and $\frac{1}{4}$ of the radius near the curved boundary. As all of the elements are approximately equilateral triangles, and are not too distorted from their master shape (another desirable feature when modelling), this gives a maximum volume difference of 25 times.

Use of an automatic free mesh generator with these element dimensions and the element type set to quadratic plane stress triangular elements produces the mesh shown in Fig. 7.4. In this mesh there are 95 elements and 220 nodes. Based on our experience of marking undergraduate assignments, good finite element results can be obtained with a 'well-constructed' mesh of around 100 elements, and so the mesh produced here should be suitable. Note, however, that while quadratic elements have the ability to fit quadratic curves, it appears from Fig. 7.4 that they are unable to model exactly the circular boundary here.

It is always good practice to solve a problem using two different finite element models as this gives some idea of the accuracy of a solution. In this case, by changing the order of the element from quadratic to linear, a second mesh is obtained that can be analysed. This is an example of the process of p-refinement, which should show that a solution is converging as the order of the element increases.

Once the mesh is built the next stage can be carried out where the numerical problem is defined on the mesh. This involves:

- Defining the load to act at the correct node, to be of magnitude $P/2$ and to be negative, to signal to the software that it acts in the negative y-direction. In this case the load at the node was set to be -0.785398 N (i.e $-\pi/4$).
- Assigning all the nodes along the quarter planes of symmetry to have the relevant displacement boundary conditions as already discussed. For each node one of its displacements is fixed and is therefore assigned a zero displacement while the other displacement is set to be free.

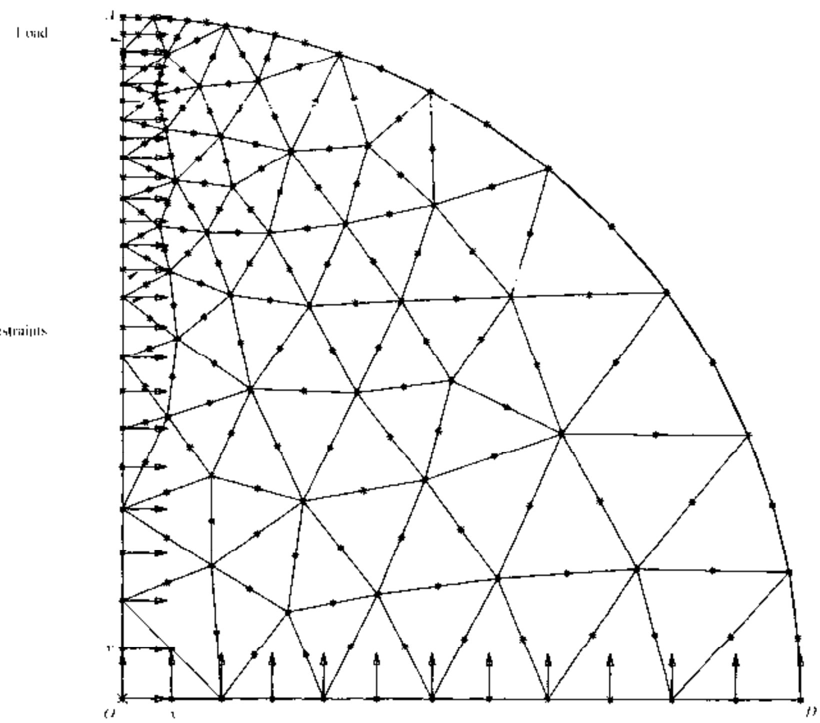


Figure 7.4 A finite element mesh with 95 quadratic plane stress triangular elements.

- Choosing default material properties of structural steel.
- Setting default physical properties with an element thickness of 1 mm.

Now the solver can be run and the results obtained. For this small model, computer time is not an issue and provided that the model is suitably restrained the solution should be generated with little trouble.

7.1.4 Finite Element Results

There are several graphical forms of output from commercial finite element software: nodal displacements, nodal stresses, and contour and vector plots of stresses. Here only the first two of these will be dealt with, as contour plots will be discussed when presenting solutions for the next example, the pressure vessel.

Figure 7.5 shows the deformed shape of the disc superimposed on the undeformed shape as calculated by the model with quadratic elements. It is clear that the deformation, and its rate of rate change with distance, is concentrated in a region below the point of loading. This provides even more evidence for the necessity of having a graded mesh. In this figure, the deformed shape is made clearly visible by multiplying the deformation by a factor of many

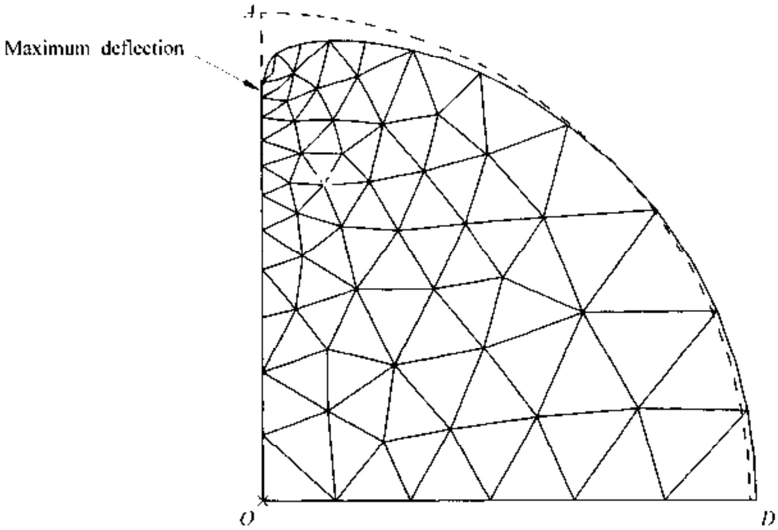


Figure 7.5 The deformed shape predicted by analysis of the quadratic element mesh.

thousands. Those not familiar with finite element analysis might mistakenly use this figure to convince themselves that the material has gross plasticity, and so it is worth emphasizing here that in a linear static analysis the software has no understanding of the strength of a material when solving (1.2). In other words, the stress results is always based on the appropriate stress–strain relationship, which for plane stress is (2.29). It is only when the modeller instructs the software to create a failure criterion contour plot, for example with a ductile metal, von Mises' stress contours might be used, that some idea as to which areas have yielded is obtained.

Profiles of finite element stress values along OA and OD are plotted in Figs 7.6–7.12, together with the classical values from (7.1) and (7.2). These stresses are shown as nodal values of σ_x, σ_y and τ_{xy} with a linear interpolation used to find the stresses at the internodal positions. This has been done for both the linear and the quadratic models. Sec. 3.7.8 stated that element stresses are calculated most accurately at the Gauss points. To obtain the nodal stresses shown, element stresses at the nodal positions have first been found from the Gauss point stresses. Hence, where a node is connected to two or more elements there are two or more stress values. To resolve this into a set of single values at each node, some form of weighting must be performed. The simplest approach is to average the element values where necessary, but this does not take account of any size disparity between connecting elements. Consequently, the software might weight the nodal values by element volume or area, which is more accurate. This aspect of nodal stress calculation can have a direct bearing on the accuracy obtained.

It is apparent from the solutions shown in Figs 7.6–7.12 that the results from the quadratic element model are superior to those from the linear element model.

There are two reasons for this. First, a quadratic element allows a linear strain and stress variation within it, while a linear element allows only constant values of strain and stress. Problems with rapidly changing strain and stress gradients are, therefore, modelled with better accuracy when quadratic elements are used. Second, as more degrees of freedom are calculated within a discrete finite element model so a better approximation is made to the actual problem, which is continuous with an infinite number of degrees of freedom. For the mesh generated here there are 95 elements with 402 active degrees of freedom in the quadratic element model but only 106 active degrees of freedom in the linear element model. If this test case is to provide a direct comparison between the performance of these two element types, then the effects of not only a p-refinement but also of an h-refinement, where the number of elements within the mesh is systematically increased, need to be compared.

Having seen that by changing the order of the element from linear to quadratic the number of variables calculated is increased by a factor of four, it may be expected that the solution time will increase substantially. In this case the actual increase in computer effort is only 25 per cent. Such an increase is not a modelling issue as the computer effort involved is so low in both cases. However, when a model requires thousands of elements and, hence, thousands of variables, such increases may be extremely significant, limiting the extent of the refinement of the mesh. In such cases analysts are advised to use fewer quadratic elements rather than a large number of linear elements.

7.1.5 Comparison of Classical and Finite Element Solutions

Now the finite element solution can be compared to the classical solution. To do this results along OA will be considered and then those along OD.

Along radii OA (y-axis) Figures 7.6–7.9 show the stresses along OA for both linear and quadratic finite element models and for the classical solution. The stress σ_x is illustrated in Fig. 7.6, with a magnified form of this being shown in Fig. 7.7. From point O for some 80 per cent of the radius the comparison between all three solutions is very good, although Fig. 7.7 shows that the linear element solution is slightly low compared to the classical solution. However, moving closer to the load point A the linear element solution departs from the classical solution quite quickly, while the quadratic element solution starts to oscillate in an attempt to stay close to the classical solution before falling rapidly negative to the physical solution at A. This difference in form of the σ_x profiles generated using linear elements, giving a smooth form, and quadratic elements, giving an oscillatory form, is typical of the behaviour for these element types in this situation and is also independent of the form of the mesh. Note that when the number of elements in a model is increased, the location along the radius where the stresses become negative moves closer to point A, but never reaches it.

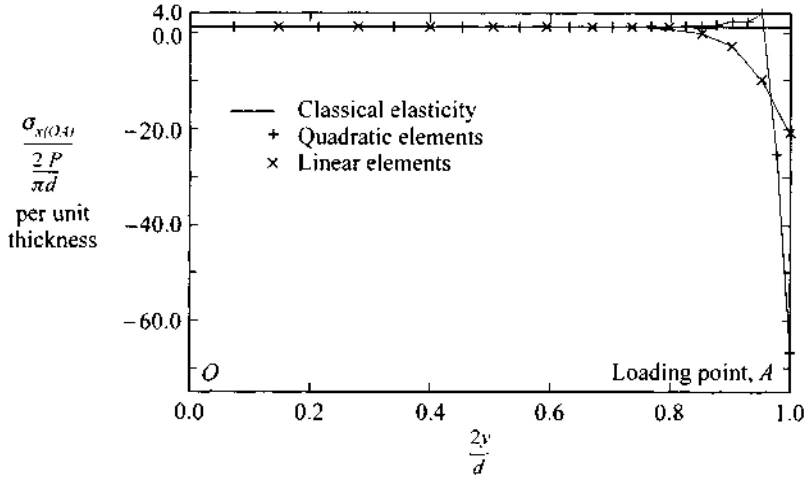


Figure 7.6 The nondimensional direct stress in the x-direction along plane OA.

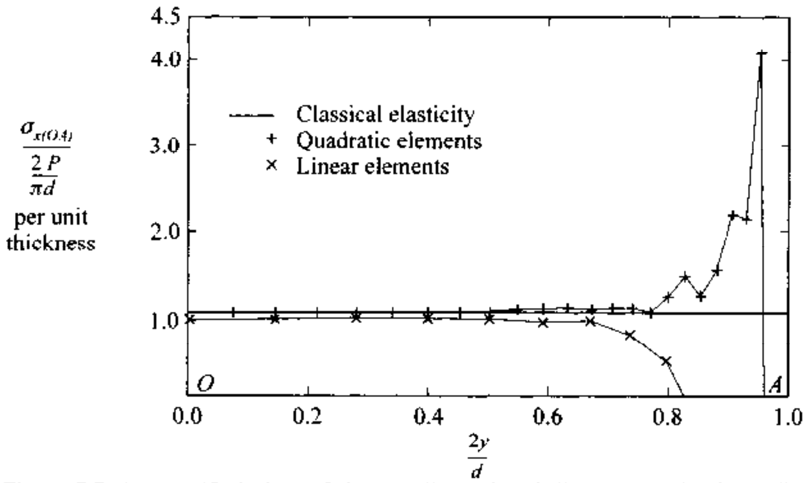


Figure 7.7 A magnified view of the nondimensional direct stress in the x-direction along OA.

Figure 7.8 shows the comparison for σ_y. The quadratic solution is seen to be quite accurate even in the region of the load point A. However, the comparison for τ_{xy} in Fig. 7.9 shows similar trends to those for σ_x.

Along radii OD (x-axis) Figures 7.10–7.12 show the stress comparisons along OD. These figures show that there is a small difference between the classical and finite element solutions when the elements are quadratic. This excellent correlation shows that the mesh construction with 6 elements, i.e. 13 degrees of freedom,

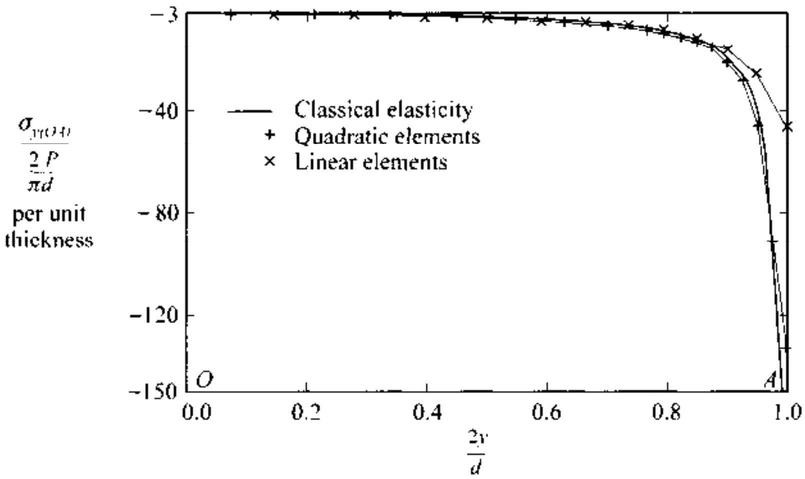


Figure 7.8 The nondimensional direct stress in the y-direction along plane OA.

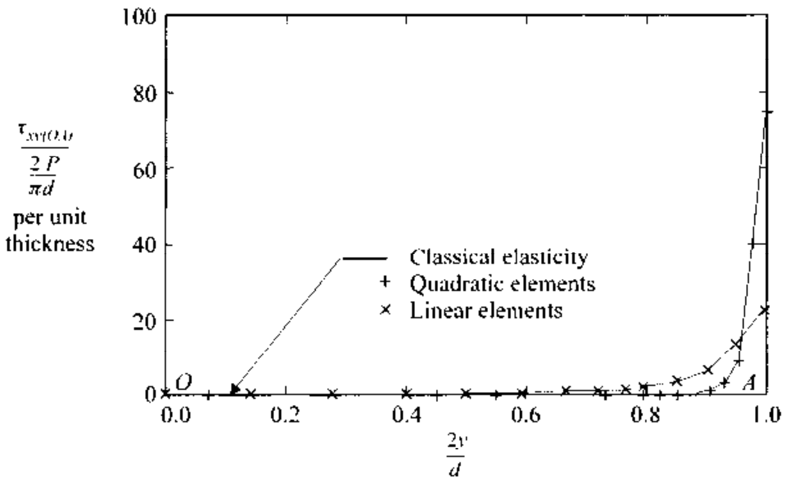


Figure 7.9 The nondimensional shear stress in the xy-plane along plane OA.

along OD is acceptable and does not require further optimization. The form of the stress profiles is similar when the element type is linear, but there is a significant error in the values of stresses σ_x and τ_{xy}. Improvement in the comparison may be obtained by increasing the number of elements in the linear element model from the present 6 elements, i.e. 7 degrees of freedom.

Clearly, the largest differences between the solutions on both axes occur along OA in the region near the load point A. To explain the behaviour of the finite element models near this point, the analyst must consider what the elements are

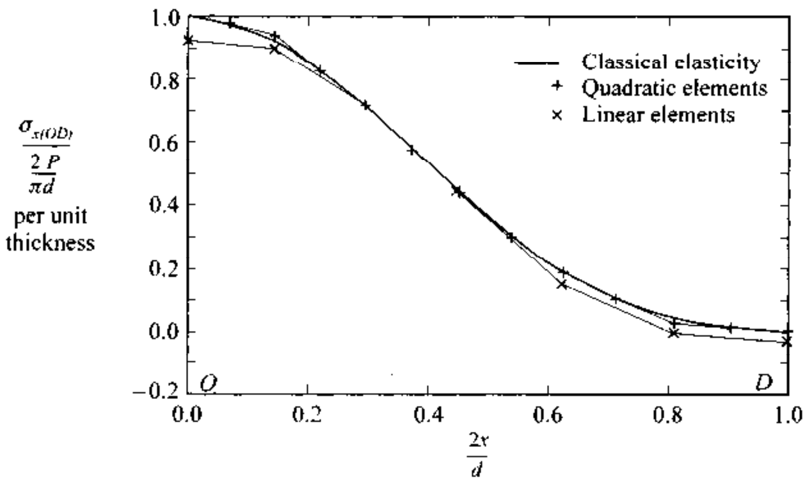


Figure 7.10 The nondimensional direct stress in the x-direction along plane OD.

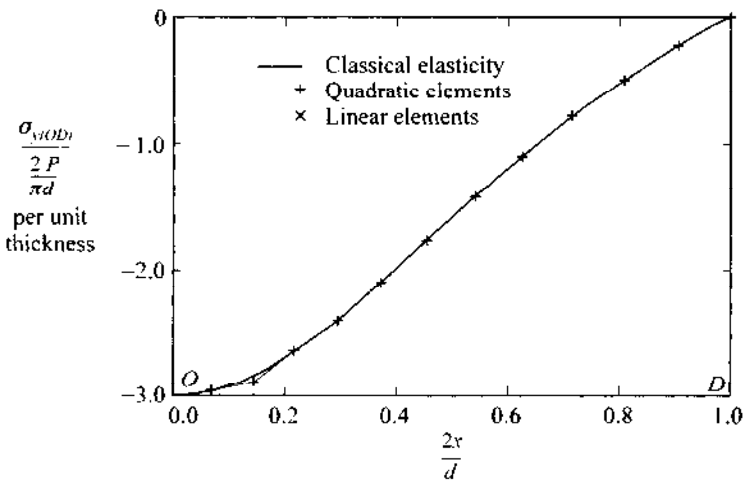


Figure 7.11 The nondimensional direct stress in the y-direction along plane OD.

actually doing. Adjacent to A the elements have the effect of averaging out the very high strain and stress gradients that occur in this region. This leads to considerable inaccuracy, so it is not too surprising to find a large difference between classical and finite element solutions in this region. The classical solution satisfies the mathematical modelling assumptions, equilibrium equations, compatibility equations and stress–strain relationships discussed in Chapter 2, and so it is exact at every point within the continuum—at least this is so for the perfect case being considered here.

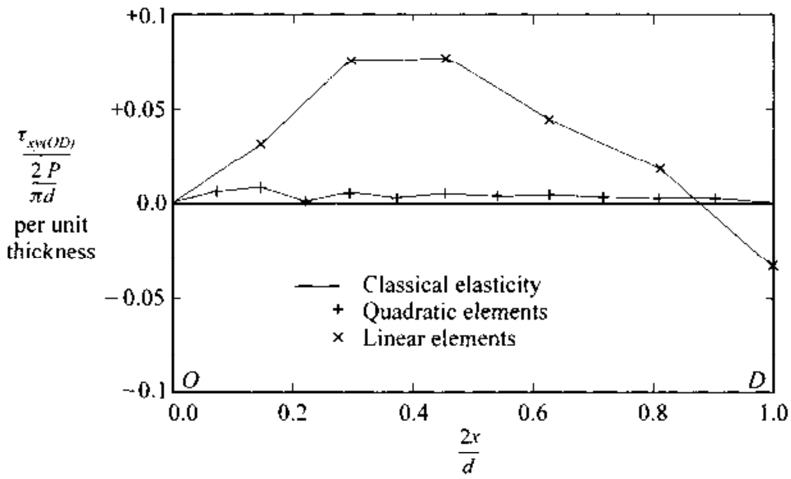


Figure 7.12 The nondimensional shear stress in the xy-plane along plane OD.

Returning to the comparison of the finite element solution with the classical solution, it is often thought that the finite element results must be wrong as they do not match the classical solution everywhere. This diagnosis is false in an engineering context, and hinges on the meaning of the word 'exact' when used in the context of a classical elasticity solution. It is a shock for many people when they are shown that the finite element solution gives a better picture of the physical reality in a practical test. There, it is found that the splitting along diameter AOB occurs only after local crushing under the loading platens. This is illustrated in Fig. 7.2 where the failure mechanism for a specimen is shown.

Thus, it is found that the finite element solution, although very similar to the classical at most points along the quarter symmetry planes, does provide valuable information as to what is happening in a real test under the point of loading.

7.1.6 Possible Improvements to the Finite Element Modelling

There are three modifications that may be made to this model to improve the accuracy of the solution:

- Using more quadratic elements in the region near to the load point.
- Rearranging of the mesh in the region adjacent to the load point such that instead of the loaded node being connected to a single element, as shown in Fig. 7.4, there are two or more elements connected to this node, as shown in Fig. 7.13.
- Distributing the load P as a surface pressure having the profile shown in Fig. 7.2. This gives a more realistic load case, but for such a surface stress distribution to be present the geometry of the disc must become flatter over the area of contact.

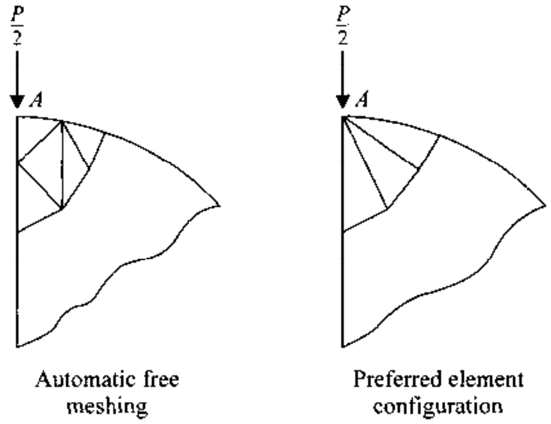


Figure 7.13 Mesh construction at the loading point.

Knight (1993) shows a variety of simple models using such techniques. Note, however, that in typical situations a concentrated force is applied at a single node. This is a common simplification for a surface pressure distributed over a small area, and if the actual surface area is smaller than the face area of elements attached to the node, then this is acceptable. However, if this causes a stress concentration that is too high, then the forces must be distributed over two or more nodes placed in such a way that the boundary stress is limited to the value of expected surface pressure.

7.2 A PRESSURE VESSEL

In this example, the analysis of a pressure vessel—an aerosol bottle—is considered. This is a real design problem that illustrates the use of general shell elements to model a bottle with a novel elliptic cross-section which has curved geometry and variable wall thickness. The bottle is made of a polymer material that is isotropic and homogeneous. In terms of modelling, the example includes the use of mapped meshing with quadrilateral elements, mirror symmetry, pressure loading and h-refinement. Difficulties occur as there are discontinuities in the geometry that give rise to poor local modelling in these regions. In particular an analysis will be made to produce a design with a suitable creep performance.

7.2.1 Problem Definition

As before, attempts must be made to build a specification for the analysis in the way that was discussed in Sec. 2.4. This means that the requirements of the analysis, the geometry of the structure, the materials and the physical loads and

restraints on the system must be defined. Again it must be emphasized that this stage of the analysis process is done before any computing takes place.

Requirements It is proposed to manufacture a domestic aerosol spray, the main container or vessel of which is to be made from a thermoplastic polymer material. At the initial design stage a finite element analysis is required to ensure that the vessel meets certain design parameters in a global sense. As will be seen from the geometry, there are discontinuities in the vessel design but for this global analysis the effects of these will be ignored to some extent. This initial analysis should show the following when the vessel body is pressurized:

- The maximum tensile stress in the vessel body, i.e. not in the complex area of the head and shoulder of the vessel, will be less than 20 N mm $^{-2}$ (MPa). If this is achieved then the creep behaviour of the polymer can be neglected and so its long-term behaviour is not a consideration. If this is the case then the materials can be assumed to have linear elastic properties, making this simple finite element analysis valid.
- The maximum distortion from the unpressurized shape at any cross-section will be not more than 1 mm, i.e. the bottle when measured by a calliper gauge will be no more than 1 mm larger across the major or minor axis when pressurized.

Geometry The thin-walled vessel, as illustrated in Fig. 7.14, consists of two injection-moulded parts welded together. These two parts are the body and the head–shoulder combination. At the top of the head section there is a flat steel cap. Nominal mid-surface dimensions and the average wall thicknesses, labelled t, are shown in Fig. 7.14. Note however, that the cross-section of the body and the shoulder of the vessel at the body–shoulder junction is elliptical with a major-to-minor axis ratio of 1.2. The shoulder cross-section undergoes transition from an elliptical section to a circular one at the shoulder–head junction, and the head is of a circular cross-section throughout. Note also that the base of the body is domed such that the dome protrudes into the volume of the vessel.

At the junctions between the dome and side walls, body and shoulder, shoulder and head and head and cap local effects occur, but these are not of concern here. These local effects are evaluated in detail using further finite element analyses after the global performance has been assured with the current analysis.

Materials The thermoplastic polymer material can be assumed to have isotropic properties with a Young's modulus of 2.5 RN mm^{-2} (GPa) and a Poisson's ratio of 0.25. It is clear that engineering experience and judgement is often needed. The material under test (coupon) is likely to have a different modulus, although this is not guaranteed, to that of the material of the vessel, and this difference could be the cause of different processing conditions. The modulus in the test is not necessarily higher, but as the FE results are used in the design, the deformation should be overestimated if a safe design is to be

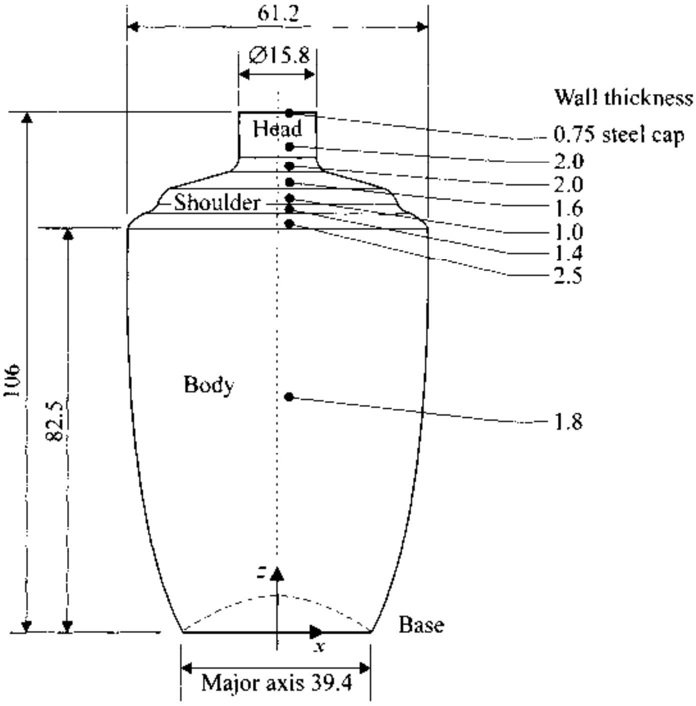


Figure 7.14 Geometry of the pressure vessel. (Dimensions in millimetres.)

ensured. There is an additional benefit in that the inherently over-stiff model that results from using finite elements formulated using assumed displacement fields, as noted in Sec. 3.5.1, is partly compensated for.

For the steel cap the standard properties for steel, i.e. a Young's modulus of 207 $\mathrm{kN\,mm^{-2}}$ and a Poisson's ratio of 0.29, can be assumed. The importance in any finite element analysis of choosing relevant and reliable material properties cannot be overstated and this issue is discussed in Sec. 2.2.6. If an analyst has a concern about any of the properties (strengths or moduli) it would always be sensible to choose values that are known to be underestimates. To choose a strength that is higher than the material can possess is to introduce a modelling error whose presence may well not be observed until the engineering product has failed.

Loads and restraints When the vessel is charged with its gas–liquid mixture, the maximum design pressure is to 0.5 $\mathrm{N\,mm^{-2}}$ (5×10^5 Pa). This pressure acts equally on all of the surfaces of the vessel. As the cross-section of the vessel is elliptical or circular, a full three-dimensional model is not required if symmetry is taken into account. If the cross-section were circular throughout then an axisymmetric model would be suitable but the elliptical cross-section leads to mirror

symmetry along only the planes of the major and minor axes. Hence quarter-symmetry can be used to allow modelling with only one quarter of the vessel. The actual boundary conditions required to achieve this will be discussed later.

7.2.2 Geometry and Mesh Definition

Before considering the creation of the computer model of the geometry and the mesh itself, both of which are fairly straightforward here, it is useful to think about the choice of element. In this case the geometry is curved and the wall thickness is very much less than the overall dimensions of the vessel. A reasonable fit to the curved geometry is achieved only if the order of the elements is quadratic or higher. Equally, it is acceptable here to use thin-shell elements, as the material is sufficiently thin and the effects of combined bending and membrane action on the displacement must be captured.

Both quadrilateral and triangular elements have been designed to give approximate solutions to Mindlin shell theory (Mindlin, 1951), where a quadratic distribution of the transverse (through-thickness) shear stress resultants is included. For curved shells, the Ahmad formulation (Ahmad, Irons and Zienkiewicz, 1970), is employed using standard eight-noded shape functions. These elements pass the patch tests for both membrane and bending actions. Note that each node has six degrees of freedom, with the displacements u, v and w being in the directions of the nodal Cartesian coordinate system axes and having dimensions of length, and the rotations θ_x, θ_y and θ_z being about the nodal coordinate system axes and measured in radians.

In Sec. 3.8.2 it was noted that there are many element formulations for general thin-shell problems, none of which exactly satisfies the requirements listed in Sec. 3.5.2 for an element to model faithfully the deformation of a continuum. As a result of this, the solutions presented here with the elements described in the previous paragraph may be quite different from those determined with another thin-shell element.

Returning to the actual geometry being considered, the data in Fig. 7.14 has been obtained from a set of engineering drawings. Note the choice of axis system where the z-axis is parallel to the centroidal axis of the vessel, the x-axis is in the direction of the major axis of the elliptical cross-section and the y-axis is in the direction of the minor axis.

As has been already mentioned, the model exhibits quarter-symmetry for the loading assumed here and so only one-quarter of the vessel need be modelled. In Fig. 7.15, the model is shown in the positive quadrant of the Cartesian system with the origin at the centre of the vessel's base.

To build the geometry the hierarchy of point to edge to surface has been used. This is done by determining key points from the engineering drawings and creating curves in the form of arcs and splines such that the 10 surfaces shown in Fig. 7.15 can each be defined from the bounding loop of curves. Note that nine of the surfaces are topologically quadrilateral and so they are suitable for the building of a mapped mesh of quadrilateral elements.

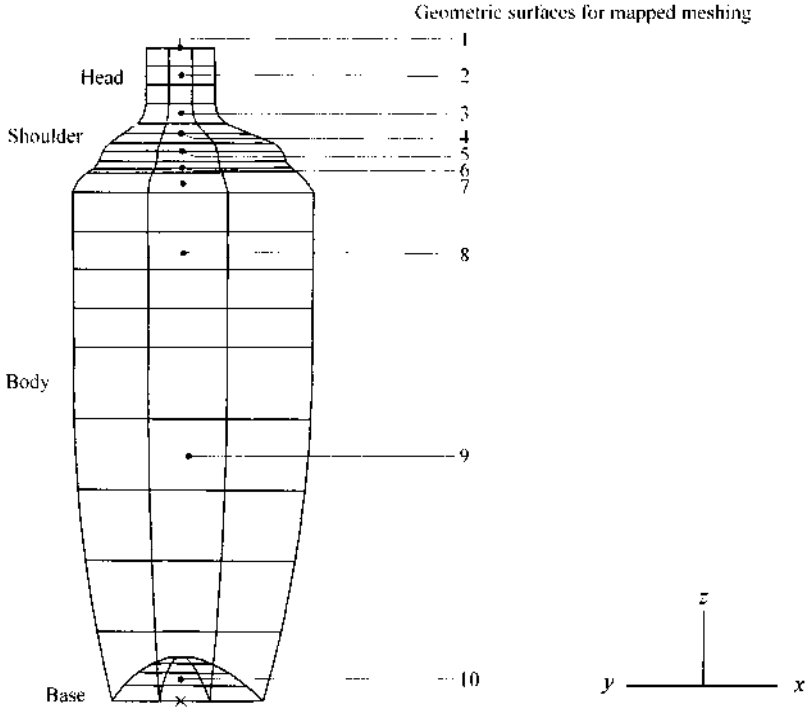


Figure 7.15 A coarse mesh of the pressure vessel.

Quadratic, quadrilateral elements, and their associated nodes, can now be placed on these surfaces, by defining the numbers of elements along each edge of a surface and the element thickness. Note that the elements within each surface have a constant thickness which is the average value of thickness for the region being considered (see Fig. 7.14). When all the surfaces have been meshed with quadrilaterals, triangular quadratic elements are created to complete the mesh where the dome of the base meets the z-axis and to model the steel cap.

A coarse mesh is shown in its complete form in Fig. 7.15 where there are three elements around the quadrant in the circumferential direction and so each element represents a surface which subtends an angle of 30°. The number of elements through the height, i.e. in the z-direction, was chosen to make the overall mesh construction acceptable. For the distribution shown, it can be seen that the distortion of the elements in the main body is not large, thereby minimizing the errors in the computation resulting from distortion of the elements from their parent shape. Note, however, that the element distortions are larger in the shoulder and head regions of the vessel. In these regions the effect of element distortion is not too important as the stress results in these regions will not be accurate because of the continually changing wall thickness.

This coarse mesh has 78 elements (72 quadrilateral and 6 triangular), 281 nodes and 1086 degrees of freedom. Given the coarse distribution of elements

some geometrical errors might be expected in the calculation. Figure 7.16 shows a section of the mesh in the transition region between the head and shoulder. The curves defining the geometry of the surfaces and the edge of the mesh can be seen to be different. Again it is clear that while quadratic elements have curved sides, they cannot, in general, fit a curved geometry exactly. The difference, however, may well be small enough to be ignored.

As a check on the convergence of the analysis, a second mesh has been constructed and this is shown in Fig. 7.17. Exactly the same methodology has been used to create this mesh but double the number of elements have been placed along the edges of each surface, and so the element density must be doubled. As this finer mesh is, effectively, the original mesh with each element split into four elements, the process of mesh refinement is known as h-refinement.

The finer mesh has 300 elements (288 quadrilateral and 12 triangular), 989 nodes and 5934 degrees of freedom.

7.2.3 Finishing the Finite Element Model

Once the mesh has been built appropriate material properties must be defined, if they have not already been defined as part of the mesh creation process. Then the correct boundary conditions are created for the model. Note that for both meshes shown in Figs. 7.15 and 7.17 the physical boundary conditions are the same, even though they are applied to different nodes and elements.

First, the internal pressure within the vessel at the design condition must be represented. In general, commercial software packages have the option to input a

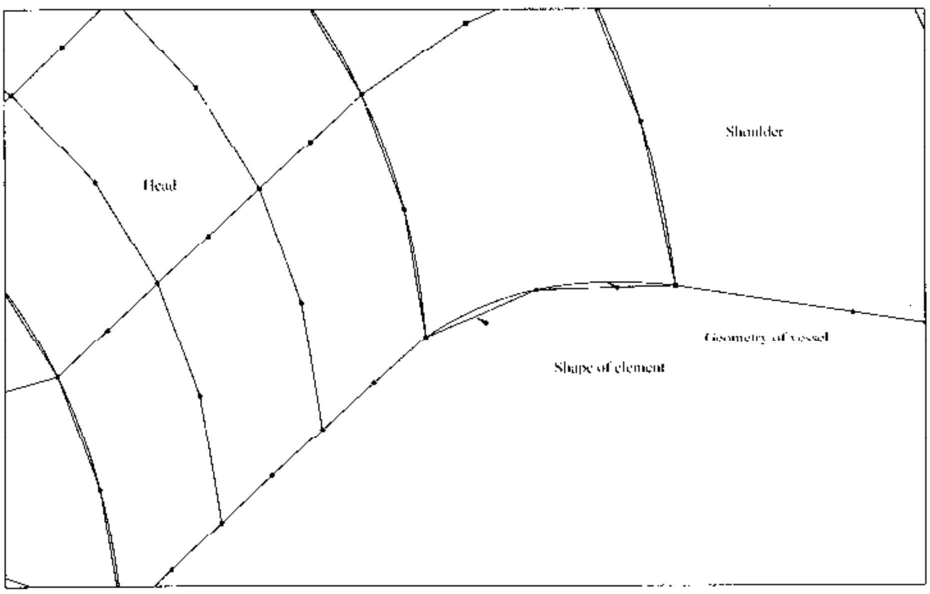


Figure 7.16 Errors in modelling the geometry of the pressure vessel.

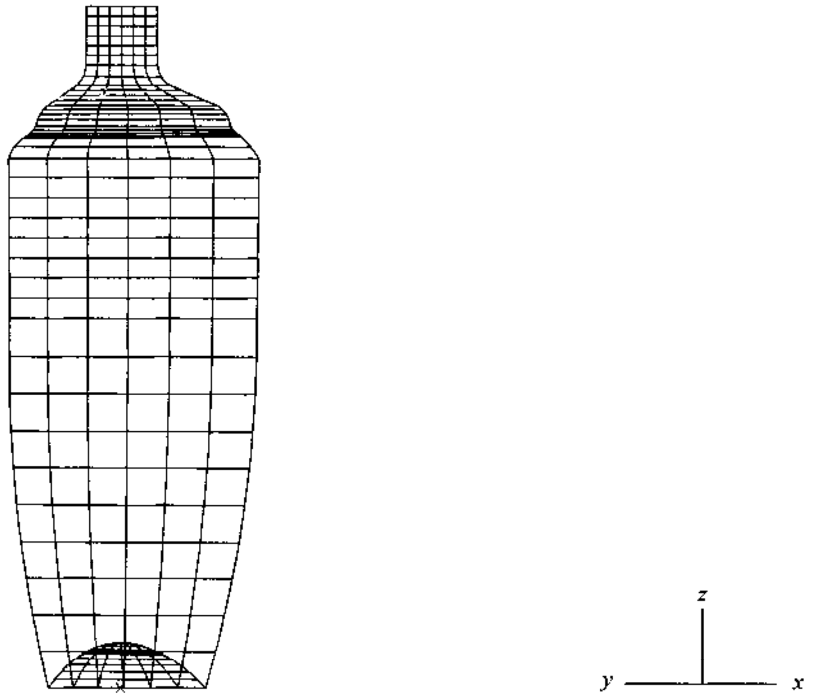


Figure 7.17 A fine mesh of the pressure vessel.

pressure loading directly. This is usually done by assigning a single value of pressure, in this case $0.5\,\mathrm{N\,mm^{-2}}$, to a series of element faces. The sign of the pressure loading defines the direction in which the pressure acts, with a positive sign denoting that the pressure acts in the directions of the positive axes of the Cartesian coordinate system. This pressure loading over the surface of an element is then converted by the software into a consistent set of nodal forces. Some loss of accuracy is introduced at this stage when the elements have curved sides and surfaces, as was discussed in Sec. 3.7.7.

To apply the pressure boundary condition all the elements are formed into a group, and for all members of the group the pressure value is set. Figure 7.18 shows the element pressure vectors on the coarse mesh, with all the vectors pointing out of the vessel, thereby indicating that the pressure within the vessel is greater than atmospheric pressure.

Displacement boundary conditions also need to be specified, to prevent rigid body motion and to enforce the correct deformation along the planes of symmetry. For a full three-dimensional model of the vessel, if the modelling of the pressure loading is exact, then the net force in all directions is zero. Equally, in this quarter-symmetric case, the overall force in the z-direction should be zero. Unfortunately, errors occur in the way in which the pressure loading is represented in the finite element analysis and, even though they are small, these errors

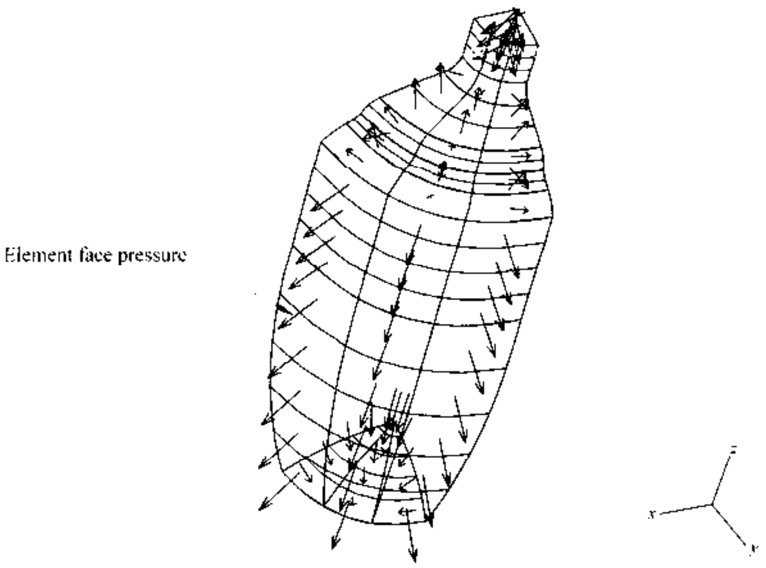


Figure 7.18 Pressure loading on the vessel.

ensure that the net integrated force due to the pressure loading it is not zero. Therefore, to restrain the vessel from moving bodily in the z-direction, at the nodes where the dome meets the body of the vessel, the w-displacement has been set to zero. This can be seen in Fig. 7.19 where a single arrowhead from a node denotes a fixed displacement.

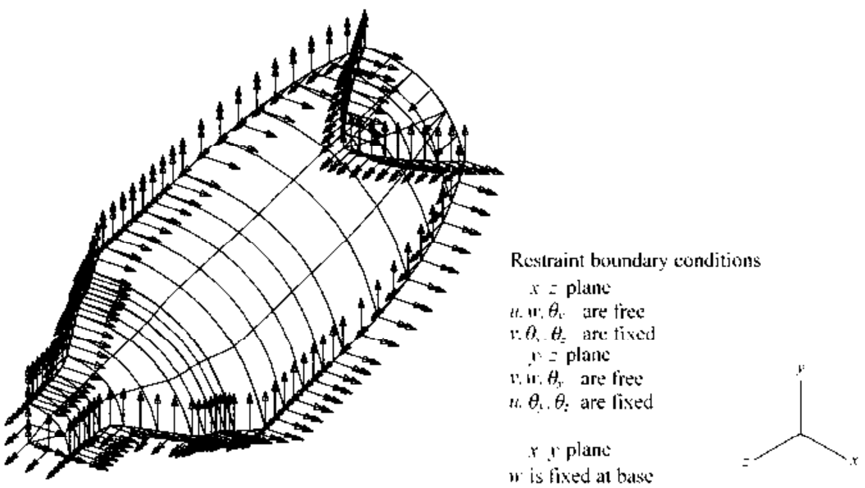


Figure 7.19 Use of restraints in setting the boundary conditions on the vessel.

Turning to the symmetry planes, let us consider a node lying along the edge of the vessel in the x–z plane. For symmetry to be enforced, these nodes must be free to move only in the x–z plane. This is done by fixing the v-displacement to be zero, shown in Fig. 7.19 by a single arrowhead again, and allowing the u- and w-displacements to be free. Also the rotation θ_y must be free and the rotations θ_x and θ_z must be fixed to be zero. These rotational boundary conditions are analogous to zero slope at the point of loading for a simply supported beam with a transverse point load. Using similar arguments for the y–z plane, the displacement boundary conditions at each node in this plane are that displacements v and w and rotation θ_x are free, and that displacement u and rotations θ_y and θ_z are fixed to be zero. Figure 7.19 shows the coarse mesh with these boundary conditions imposed. Note the use of a double arrowhead to denote that the rotation about the axis direction has been fully restrained.

Now the computer model is complete and the solver can be run to produce the required results.

7.2.4 Finite Element Results

From our understanding of the requirements of this analysis, the results must be examined to determine the displacement of the vessel under the pressure loading and the maximum tensile stress within the material. So that any output can be easily visualized in the context of the vessel's geometry, the displacement results are presented for the deformed and undeformed shapes on one figure, and the stress results are given on the undeformed mesh. Analysts should not underestimate the time that it takes to generate the appropriate graphical pictures from the numerical data when using post-processors, as complex operations are often required to obtain, for example, a view of the geometry in the correct orientation or the correct levels for stress contours.

Figures 7.20 and 7.21 show the outline of the deformed shape together with the original coarse mesh, and fine mesh, respectively. Note that the deformed geometry is calculated from the magnitude and direction of the deflection at each node in the mesh, and that in these figures different magnifications have been used to show the deformed geometry. Without magnification the displacement from the original shape cannot be seen. From these figures the predicted deformation shows that the dome and the cap are forced outwards and that the shape expands along minor axis while contracting along the major axis. Hence the vessel becomess more circular in cross-section. This ties in with engineering intuition, which suggests that a thin-walled tube of elliptical cross-section would becomes more circular under a pressure loading, as energy arguments can be used to show that a circular section is the ideal shape for pressure loading. This also acts as a check on the modelling of the pressure vessel.

Figures 7.20 and 7.21 also show the maximum deflections in each of the Cartesian coordinate directions, together with the locations at which these deflections occur. By comparing these values and the overall deformation of each mesh, it is clear that the two meshes perform very similarly, with an error of

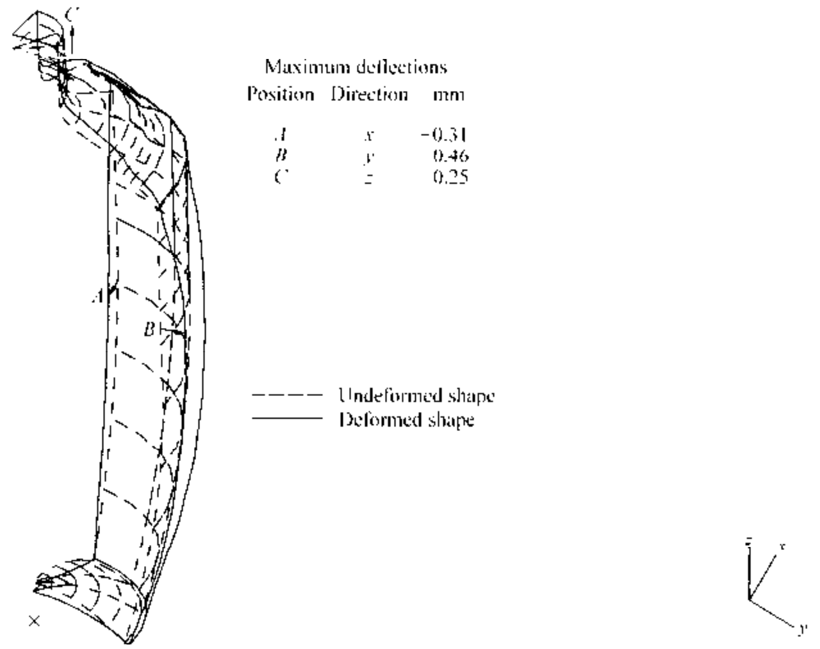


Figure 7.20 The displaced shape predicted with the coarse mesh.

approximately 3 per cent. Hence the h-refinement has shown that the original coarse mesh is suitable to this level of accuracy.

Note that the maximum deflection of the vessel body is 0.47 mm at point *B*, and is calculated with the fine mesh (Fig. 7.21). Owing to the restraints that have been applied, this means that the bottle increases in size by twice this value, i.e. 0.94 mm. As this is less than the specified limit for the increase of 1.0 mm, it can be concluded that the stiffness of the vessel is acceptable.

To consider the stresses within the model, the usual tool is a contour plot displaying a range of stress levels. In examining these plots the analyst can make some simple, yet valuable, checks. For example, the boundary conditions of the vessel require that the stress component normal to the surface must be equal to the pressure loading. Equally, at the planes of symmetry, the contour must be normal to the symmetry boundary. In general, there should be no abrupt changes in the direction of a contour and the contour should be continuous. However, where there is a change in material thickness, which in the vessel occurs at the boundaries of the 10 surfaces that were meshed, or where the geometry has a discontinuity, e.g. at the dome to side wall and body to shoulder interfaces, it is to be expected that the contour plots will be neither smooth nor continuous. Also, the contours should show the peak values of stress in the expected places.

Be aware that software packages do not always acknowledge the method used to calculate the nodal stress values which are then interpolated to create the

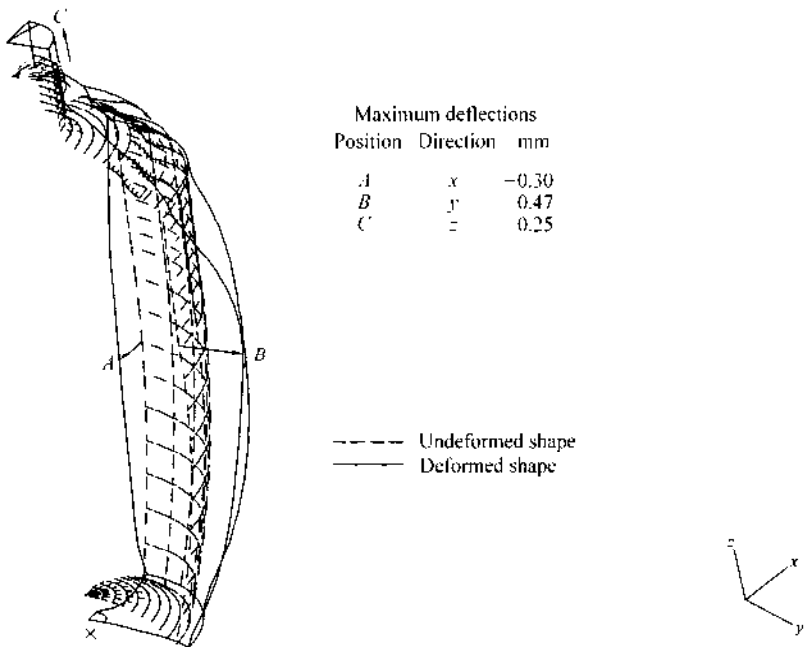
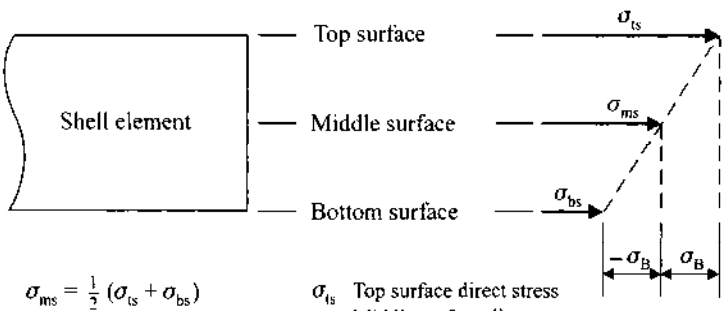


Figure 7.21 The displaced shape predicted with the fine mesh.

contour plots. For this reason these plots should never be used to find an accurate value of the peak stresses. If such values are required then the raw data, probably at Gauss points or possibly at the nodes, must be interrogated directly.

When generating a contour plot a commercial package may use default settings or an automatic data scaling routine to decide the intervals between each contour. Such an algorithm scans the nodal values to find the data range and then selects the contour intervals to cover the range of values for a given number of contour lines. Often the analyst has to override these defaults and set the range and intervals manually. From experience, plots with between 6 and 10 intervals give the best compromise between information provision and the clarity of display. If more than 10 intervals are used there is a danger that the plots become cluttered and difficult to follow. Equally, if fewer than 6 contours are used not enough information is presented and major features of the stress distribution may be missed.

Before presenting the stress contours for the vessel, the type of stress output available for a thin-shell element must be explained. Figure 7.22 shows that there are four direct stress values available. The element deforms due to both membrane (i.e. the middle surface stress σ_{ms}) and bending (i.e. the bending stress σ_B) actions. Note that for the isotropic shell element the bending behaviour is that of pure bending (see Sec. 2.3.1), where normals remain straight and normal. Combining the membrane and bending stresses gives two surface stresses, σ_{ts} at the top and σ_{bs} at the bottom.



$$\sigma_{ms} = \tfrac{1}{2}(\sigma_{ts} + \sigma_{bs})$$

$$\sigma_B = \tfrac{1}{2}(\sigma_{ts} - \sigma_{bs})$$

σ_{ts} Top surface direct stress
σ_{ms} Middle surface direct stress
σ_{bs} Bottom surface direct stress
σ_B Bending direct stress

Figure 7.22 Bending stresses within shell elements.

As stresses are referred to the global Cartesian coordinate system, when selecting a principal stress (maximum, middle or minimum) the post-processor may calculate the three principal values from the displacement data. Values are determined to represent the top and bottom surfaces, as the post-processor assumes that the stress on the mid-plane is an average of the stress at the top and bottom surfaces.

Figures 7.23 to 7.31 are contour plots of the maximum principal stress, with the results for the coarse mesh being given in Figs 7.23–7.28 and those for the fine mesh in Figs. 7.29–7.30. Each plot gives the peak tensile and compressive stress, and the contour interval has been set manually to 5 N mm^{-2}. Note that the default settings in packages may lead to stress values for a contour being given to seven significant figures. Such use of output formatting could fool an inexperienced analyst into believing that this is also the accuracy for the stress output. The various procedures used by the post-processor generate contours from the raw data. This is known to introduce, in addition to those already inherent in the finite element analysis, further numerical errors. Hence, contour levels should not be given to more than three significant figures. In fact, in most cases, it is advisable to use only two significant figures.

From our understanding of the requirements of this analysis, if creep behaviour is to be neglected in design, then the direct stress must be less than 20 N mm^{-2} in the body section of the vessel. Furthermore, it is expected that the peak values will be at the outer surfaces of the shell because of the combined membrane and bending actions, and so Figs 7.23 and 7.24 are plots of stresses σ_{ts} and σ_{bs} for all the elements in the model. From these figures it can be seen that the contours are not smooth or continuous in the regions of the vessel where the thickness changes or there are discontinuities in the shape of the vessel. Away from these regions, in the body of the vessel, it is found that the contours are fairly smooth with only slight rippling. It is only in such areas that the predicted stress

Contour	Direct stress σ_1, N mm^{-2}
1	−5
2	0
3	5
4	10
5	15
6	20
7	25
8	30

Peak values	
Compressive	39
Tensile	30

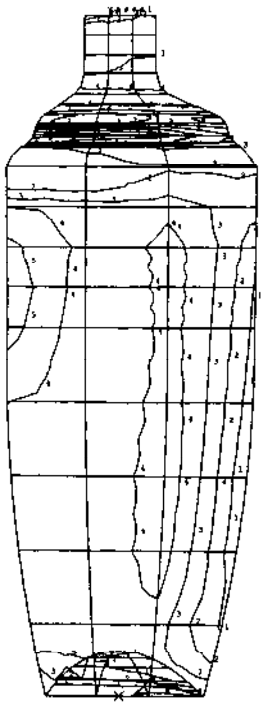


Figure 7.23 Maximum principal stress at the top surface (coarse mesh).

values can be used reliably, and inspection of the stress contours in the body region, away from discontinuities, shows that the surface stresses only just exceed the limiting stress of 20 N mm^{-2}.

To remove the influence on the generation of the contour plot of the dome, the shoulder and head of the vessel, the elements in these regions are not included in the post-processing for Figs 7.25 and 7.26. In Fig. 7.25 the stress at the top surface is shown, and so this figure can be compared to Fig. 7.23. It can be seen that the contours have changed only close to where elements have been removed. By limiting the stress contour plots to only those elements where the material thickness is constant, the effect of geometry changes is minimized. This same procedure of isolating specific parts of the structure when producing contour stress plots is also carried out if there are dissimilar material types or an abrupt change in the structural loading, for the same reasons.

When comparing Figs 7.25 and 7.26 to Figs 7.23 and 7.24, it can be seen that in the middle region of the body there is hardly any difference. Here, Saint-Venant's principle is at work and the 'die-away' length for this example is several times the wall thickness of 1.8 mm. For the reasons already discussed, the stress contours are not expected to be accurate close to the two interfaces and so the high stress values there should be considered as inaccurate.

Contour	Direct stress σ_{bs} N mm^{-2}
1	0
2	5
3	10
4	15
5	20
6	25
7	30
8	35

Peak values	
Compressive	31
Tensile	62

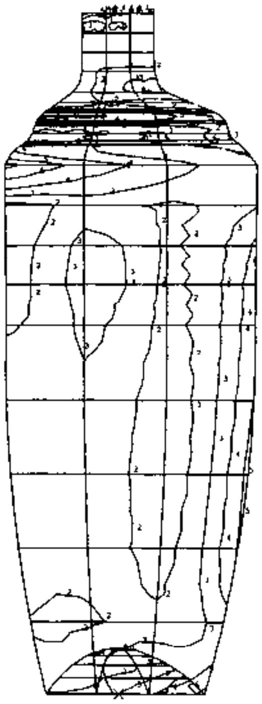


Figure 7.24 Maximum principal stress at the bottom surface (coarse mesh).

Contour	Direct stress σ_{ts} N mm^{-2}
1	−5
2	0
3	5
4	10
5	15

Peak value	
Tensile	19

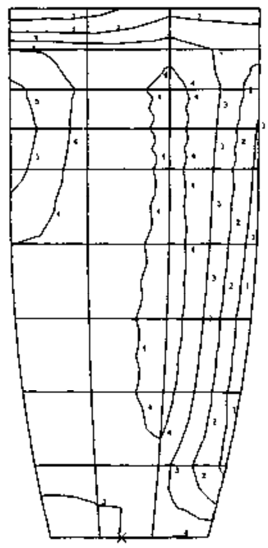


Figure 7.25 Maximum principal stress at the top surface—body only (coarse mesh).

Contour	Direct stress σ_{bs} N mm^{-2}
1	5
2	10
3	15
4	20
5	25
6	30
Peak value	
Tensile	34

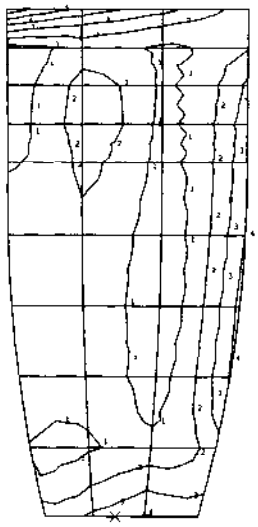


Figure 7.26 Maximum principal stress at the bottom surface—body only (coarse mesh).

New information on the stress distribution is provided in Figs 7.27 and 7.28. In Fig. 7.27 the middle surface stress σ_{ms}, known as the membrane stress, is fairly constant within the body having a value between 5 and 10 N mm^{-2}. In contrast, from Fig. 7.28, the bending stress σ_B continually varies from values below -5 N mm^{-2} to values greater than 25 N mm^{-2}. This shows that for the vessel the bending action is dominant over the membrane action.

Contour	Direct stress σ_{ms} N mm^{-2}
1	5
2	10
Peak value	
Tensile	13

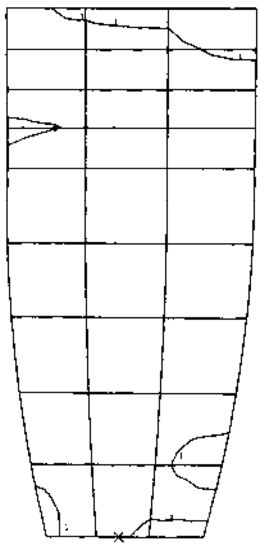


Figure 7.27 Maximum principal stress at the middle surface—body only (coarse mesh).

Contour	Direct stress σ_θ N mm^{-2}
1	−5
2	0
3	5
4	10
5	15
6	20
7	25

Peak values	
Tensile	28
Compressive	8

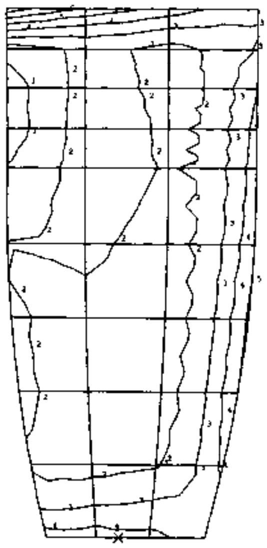


Figure 7.28 Maximum principal stress due to bending—body only (coarse mesh).

Stress results for the fine mesh are shown in Figs. 7.29 to 7.31. By comparing these figures with their coarse mesh equivalents it can be seen that, if anything, the contours are less smooth and have more discontinuities. It is, however, reassuring to find that the peak stress values are little different. This indicates that it is not necessary to run a third model with a mesh which has been generated by further h-refinement. Hence the results generated with both the coarse and the fine mesh models can be taken as being representative. This says nothing, however, about the worthiness of the results with regard to what actually happens.

7.2.5 Further Modelling Possibilities

The specification for the design of the vessel has been limited to the serviceability states of limiting maximum deformation and working tensile stresses due to the vessel's pressure charge. Analyses are also needed to model such things as impact behaviour. For example, a critical situation can be identified when a free-falling vessel impacts a hard surface and the point of contact is the interface between head and body. A linear elastic small displacement analysis can still be used if the impact load is modelled as an equivalent static load. From experience, such an impact load can be modelled as a static load which acts normal to the point of contact and the magnitude of which is twice the dead weight of the vessel. As a consequence of this the models defined here can be used providing that the boundary conditions are modified.

The causes of ultimate failure of the vessel may also need to be known. To do this a full nonlinear analysis must be carried out as the polymer material does not

Contour	Direct stress σ_{1s} N mm^{-2}
1	−5
2	0
3	5
4	10
5	15
6	20
7	25

Peak values

Tensile	30
Compressive	38

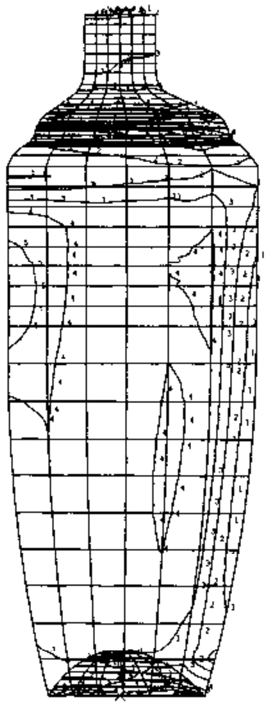


Figure 7.29 Maximum principal stress at the top surface (fine mesh).

Contour	Direct stress σ_{1s} N mm^{-2}
1	−5
2	0
3	5
4	10
5	15

Peak values

Tensile	20
Compressive	10

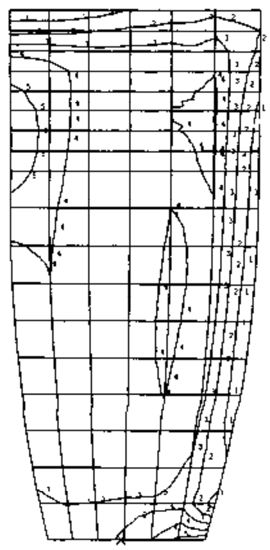


Figure 7.30 Maximum principal stress at the top surface—body only (fine mesh).

Contour	Direct stress σ_{bx} N mm^{-2}
1	5
2	10
3	15
4	20
5	25
6	30
7	35
Peak value	
Tensile	37

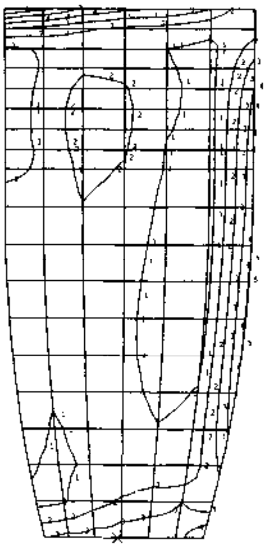


Figure 7.31 Maximum principal stress at the bottom surface—body only (fine mesh).

behave in a linear elastic way when failure occurs, as discussed in Sec. 2.2.6, as the peak deformations exceed the wall thickness and material properties are nonlinear. It may even be necessary to carry out a full nonlinear analysis for the impact condition if the deformations and strain rates are found to be large.

In all these nonlinear cases, global models may be used initially to identify the locations where failure is likely to occur. If such locations are near to discontinuities in the geometry or thickness, then a shell model will not provide accurate results that can be used to determine the loading causing failure. The analyst then has to decide if a model of local regions of the vessel is necessary. This may be achieved using the substructuring technique, explained in the frame example in Chapter 8, where the three-dimensional solid geometry of a local region is modelled using boundary conditions from the results of a global model. This is the only technique that can be used to determine the stress variations at the discontinuities in a shell model. However, it is not the panacea that the designer desires unless the exact material property variation and exact boundary conditions are known and can be modelled.

So it can be seen that the designer constantly wants the most accurate finite element results from the resources available, in terms of funding, the analysis team, software and hardware.