THEORETICAL GUIDE

Version 3.0 beta

Concept Analyst Ltd.
Durham University
Department of Engineering
South Road
Durham DH1 3LE

June 2020

Contents

1

Introduction

Concept Analyst is a Windows-based software system for the calculation of information that allows engineers to assess the durability of mechanical and structural components. It performs these calculations in a sketching environment that allows very rapid solution of the stress distributions through general, two-dimensional shapes. A re-analysis capability allows many design changes to be made while the program automatically updates the stresses that are displayed on the screen.

The program performs numerical analysis calculations that have long been available in commercial software systems. Namely, it performs linear static stress analysis. The advantage that Concept Analyst offers over these well-established packages is the simplicity and speed of usage.

While traditional stress analysis software is based on the Finite Element Method (FEM), Concept Analyst uses a different technique called the Boundary Element Method (BEM). This is fundamental to the speed of model creation and flexibility of the model changing and re-analysis included in the program.

This guide presents a theoretical overview of the BEM and its implementation in Concept Analyst.

2

Elasticity theory

2.1 Basic definitions

An introduction to the concepts and equations of linear elasticity is provided here to make the engineer aware of the meaning of the results displayed by Concept Analyst. This is important since some quantities are subject to being defined in different ways in different books and by different authors.

Concept Analyst uses a conventional, Cartesian coordinate system in which to define the geometry, boundary conditions, displacements and stresses. The origin of this (x,y) system is not displayed on the screen, nor are the axis directions, but it will always be assumed that the x-direction is positive to the right and the y-direction is positive upwards.

In this coordinate system we define the following quantities:

x: the horizontal coordinate of a point
y: the vertical coordinate of a point
u: the displacement in the x-direction
v: the displacement in the y-direction
In addition we define a set of stresses with the implicit sign convention that:

- tensile stress shall be positive
- compressive stress shall be negative
- shear stress shall be positive if aligned as shown in figure 1.

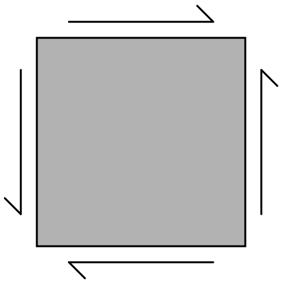

Figure 1. Shear stress directions deemed positive in the sign convention

Within this sign convention, the notation to be used will be as follows:

σ_x: the direct stress in the x-direction
σ_y: the direct stress in the y-direction
τ_{xy}: the shear stress in the xy-plane

Strain components will be denoted in the usual way. The same sign convention will be adopted as for stress, such that tensile direct strains (ε) will be positive, and shear strains (γ) caused by a set of shear stresses as shown in figure 1 will also be positive.

ε_x: the direct strain in the x-direction
ε_y: the direct strain in the y-direction
γ_{xy}: the shear strain in the xy-plane

In this two-dimensional system, some condition needs to be assumed for the behaviour in the third direction, i.e. perpendicular to the xy-plane in which the model is defined. Denoting the third direction 'z', it is customary to define one of the following:

- Plane stress, in which the stress $\sigma_z = 0$
- Plane strain, in which the stress $\varepsilon_z = 0$

Typically, plane stress is applicable for the analysis of thin sheets and plates. Plane strain might be used for the analysis of a problem which is two-dimensional because of some prismatic nature of the geometry and boundary conditions.

The default condition for Concept Analyst is plane stress.

2.2 Stress-strain relations

The stresses and strains in the xy plane may be related by simple concepts of elasticity.

If we assume plane stress conditions ($\sigma_z = 0$) then the direct strain in the x-direction, ε_x, is caused by the actions of both σ_x and σ_y. We consider both independently and add the strains using the principle of superposition. If we adopt the usual notation ν for Poisson's ratio and E for Young's modulus, we can write the effects of the stresses on the strain ε_x as:

$$\varepsilon_x = \frac{\sigma_x}{E}$$

$$\varepsilon_y = \frac{-\nu\sigma_y}{E}$$

and by the principle of superposition

$$\varepsilon_x = \frac{1}{E}(\sigma_x - \nu\sigma_y)$$

and similarly for the strain in the y-direction

$$\varepsilon_y = \frac{1}{E}(\sigma_y - \nu\sigma_x)$$

After some algebra, these equations can be written to give the stress components in terms of the strains:

$$\sigma_x = \frac{E}{1-\nu^2}(\varepsilon_x + \nu\varepsilon_y)$$

$$\sigma_y = \frac{E}{1-\nu^2}(\varepsilon_y + \nu\varepsilon_x)$$

In plane strain, similar reasoning allows the following relationships to apply:

$$\sigma_x = \frac{E}{(1-2\nu)(1+\nu)}\left[(1-\nu)\varepsilon_x + \nu\varepsilon_y\right]$$

$$\sigma_y = \frac{E}{(1-2\nu)(1+\nu)}\left[(1-\nu)\varepsilon_y + \nu\varepsilon_x\right]$$

$$\sigma_z = \nu(\sigma_x + \sigma_y)$$

2.3 Stress transformations

While knowledge of the direct and shear stress components σ_x, σ_y and τ_{xy} allows a complete picture to be formed of the stress conditions at a point, they do not immediately provide the engineer with the full information to assess durability. In particular, if the shear stress τ_{xy} is non-zero, then there will exist stresses in other directions in the plane that are larger than either σ_x or σ_y.

By considering the force equilibrium of a piece of material cut at an angle θ to the vertical, we may derive a set of equations giving the direct stress in a direction rotated through this angle. This set of equations is most conveniently expressed using Mohr's circle for stress, which is shown in figure 2.

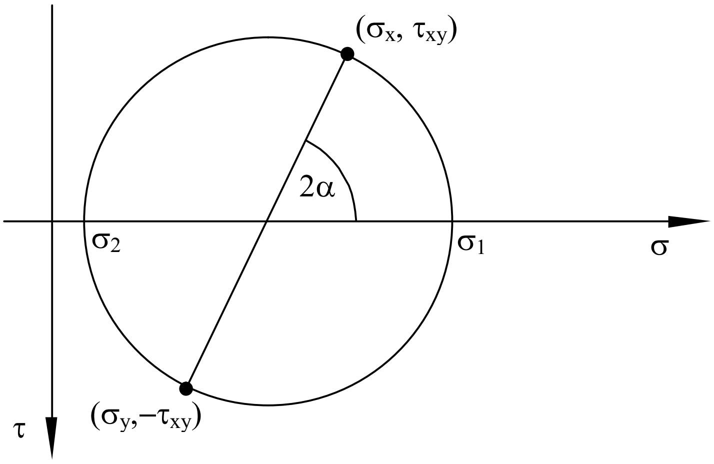

Figure 2. Mohr's circle for stress (example)

Notice some key elements of Mohr's circle:

- The vertical axis is plotted with positive shear stresses in the downward direction.
- The points (σ_x, τ_{xy}) and $(\sigma_x, -\tau_{xy})$ both lie on the circumference at opposite ends of a diameter.
- Rotation in the Mohr's circle through an angle 2θ represents rotation in the real material through angle θ in the same direction[1]

The stress σ_1 is the most tensile stress experienced in any direction in the xy plane at the point. In the example of figure 1, this stress would act in a direction rotated clockwise through angle α from the direction of σ_x.

[1] It is for this reason that the vertical axis is plotted unconventionally with positive shear stresses in the downward direction. If the vertical axis were plotted with positive upwards, the rotation in the Mohr's circle would be in the opposite direction to that in the real material. Many authors adopt this alternative convention.

The stress σ_2 is the most compressive stress in the xy plane. Notice that, as in figure 1, this may actually be a tensile stress if there is no compression in any direction. Likewise, of course, if there is no tension in the plane, then the stress σ_1 may be compressive.

The stresses σ_1 and σ_2 are called **principal stresses**.

There are always three principal stresses that act at a point. In a two-dimensional case, the third principal stress will be:

- $\sigma_3 = 0$ in a plane stress case
- $\sigma_3 = \nu(\sigma_x + \sigma_y)$ in a plane strain case

Concept Analyst makes the following definitions with respect to the three principal stresses:

- The most positive principal stress is termed the ***maximum principal stress***.
- The most negative principal stress is termed the ***minimum principal stress***.
- The third principal stress is termed the ***middle principal stress***.

Notice that these three definitions make no assumptions about whether any of these stresses are positive or negative, i.e. tensile or compressive.

2.4 Failure criteria

Knowledge of the principal stresses allows the engineer to relate the stress conditions directly to the material properties to determine the likelihood of yielding failure. For metals, this is commonly determined on the basis of one or more of the following criteria:

2.4.1 Maximum principal stress criterion

If the maximum principal stress exceeds the tensile yield stress of the material, then failure is deemed to occur. Likewise, failure will occur if the minimum principal stress is a compressive stress that exceeds the yield stress in compression.

2.4.2 Tresca criterion

If the difference between the maximum and minimum principal stresses exceeds the material's yield stress then failure is deemed to occur. This is derived from the relationship between the maximum shear stress and the yield stress.

2.4.3 Von Mises criterion

The Von Mises failure criterion is derived from the shear strain energy in a material, i.e. the strain energy associated with change in shape and not that associated with the change in volume. Failure will be deemed to occur if the expression for what has become known as the 'Von Mises stress', here denoted σ_{VM}

$$\sigma_{\mathrm{VM}} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}$$

exceeds the material's yield stress. Thus, the Von Mises stress is related to the root mean square of the three Mohr's circle diameters in a three-dimensional stress field.

The Von Mises stress is a useful measure since it not only provides a failure criterion that is reasonably generally applicable for ductile materials, but it is also formulated as a single stress value (σ_{VM}) that is positive for both tension and compression and can be plotted in contour form or in some other easily accessible manner.

3

The Boundary Element Method

3.1 Introduction

The stresses and displacements produced by Concept Analyst are found using a mathematical technique called the Boundary Element Method (BEM). This is similar in many ways to the more common Finite Element Method (FEM), which is used by several major commercial stress analysis systems. Like the FEM, the BEM works by dividing the object into small parts, and by developing a large set of simultaneous equations relating the stresses and displacements in the various parts of the component being analysed. However, there are some differences that are quite fundamental. This section of the Theoretical Guide introduces the main ideas behind the method.

For the user, the principal difference between the FEM and the BEM is this – the FEM requires the volume of the object being analysed to be divided into volumetric elements, whereas the BEM requires only the boundary of the object to be divided into elements. This means that, for the two-dimensional geometries considered by Concept Analyst, only the perimeter of the object needs to be defined, leaving the enclosed area empty. This can be clearly shown if the 'Draw elements' check box is set in the 'Edit – Preferences…' dialog box. After an analysis is run the elements are shown, such as those illustrated in figure 3.

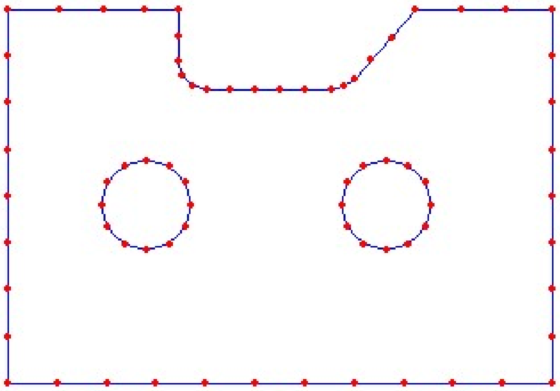

Figure 3. Typical boundary element 'mesh'

In developing the BEM theory in stress analysis in this section, we will be bearing in mind this conceptual difference between the FEM and BEM.

The BEM may be applied to a wide range of engineering problems that are governed by various differential equations. Examples include heat transfer, acoustics, corrosion modelling and electromagnetics. However, the method is not so advanced as the FEM in terms of the range of analysis types available, so that commercial BEM software does not extend greatly into problems involving time dependency, non-linearity or other forms of varying material properties.

In a nutshell, then, the FEM and BEM may be compared simplistically as follows, at least in the commercial software environment:

- The FEM is more versatile

- The BEM is easier to use

Concept Analyst has been developed primarily as an easy-to-use analysis tool. It is for this reason that the BEM has been adopted. The technique is also very well suited to the re-analysis capability offered by Concept Analyst.

3.2 Units

The units used in the analysis do not matter. The only important fact is that they should form a consistent set. For example,

- Newtons, metres, Pascals
- Newtons, millimetres, megaPascals
- Pounds force, inches, psi

Concept Analyst defaults to the use of Newtons and millimetres, so that the unit of stress or Young’s modulus will be N/mm^2. This is equivalent to writing stress or Young’s modulus in MPa.

3.3 The reciprocal theorem

The idea of the reciprocal nature of structural systems is well known in mechanics. There are numerous ways of expressing the ***reciprocal theorem***, one of which is given here as a starting point in the development of the BEM for elasticity.

Imagine we have an object and we apply to it two different load cases. Load case A contains some forces and displacement constraints. Load case B consists of a different set of forces and displacement constraints. The reciprocal theorem states that the work done by the forces from load case A on the displacements from load case B is equal to the work done by the forces from load case B on the displacements from load case A. That is,

$$\text{Forces}_A \text{ x } \text{Displacements}_B = \text{Forces}_B \text{ x } \text{Displacements}_A$$

Engineers familiar with the principle of Virtual Work will recognise this type of formulation of the problem. This works for any two arbitrary load cases.

Let us now describe our load case A in a little more detail, by stating that it consists of a set of boundary tractions ‘t’, displacements ‘u’ and body forces ‘b’. Defining terms, ‘traction’ is a very useful quantity. It is like a surface stress, so that a traction applied perpendicular to the surface would be a distributed load (or pressure) applied in this direction, and a traction applied tangentially to the surface would be a shear stress. So traction has the same units as stress. Traction also has another very useful property that stress does not share – we can resolve tractions into their orthogonal components or, in reverse, think in terms of a resultant traction. This is very useful in applying traction boundary conditions on inclined and curved surfaces.

The body forces ‘b’ might include, for example, gravitational loads, thermal loads in a stress analysis or centrifugal loads.

We will also state at this stage that load case A will be the real load case that we are analysing. Load case B is completely arbitrary at this stage, but let us also describe load case B in more detail, by stating that it consists of tractions ‘t*’, displacements ‘u*’ and body forces ‘b*’.

The work done by the surface tractions and body forces on a set of displacements may be found by integrating over the component, so that the reciprocal theorem may be stated

$$\int_V t^* u dV + \int_V b^* u dV = \int_V u^* t dV + \int_V u^* b dV \tag{1}$$

where V is the volume of the object being analysed. Notice that if we have integrals over the volume we will require elements through the volume (finite elements). To proceed with boundary elements we need to remove these volume integrals and replace them with integrals only over the surface.

3.4 The boundary integral equation

The first methods of removing the volume integrals in equation (1) are simple. Firstly we recognise that the tractions t and t* apply only on the surface S, and that there are no tractions anywhere else in the volume. This means that the volume integrals containing tractions t and t* may be rewritten as surface integrals giving

$$\int_S t^* u dS + \int_V b^* u dV = \int_S u^* t dS + \int_V u^* b dV \tag{2}$$

Secondly, we will state for simplicity that there are no body forces acting in the real load case (A), so that b = 0 everywhere in the component. We remove the volume integral containing body force ‘b’ leaving

$$\int_S t^* u dS + \int_V b^* u dV = \int_S u^* t dS \tag{3}$$

which contains only one volume integral. Body forces can be considered in the BEM, for example as the self-weight option in Concept Analyst, and this will be considered later in this guide but for now we neglect body forces for simplicity.

To remove the last volume integral from the equation, we have to start to be a little more creative. In fact, we will do this by starting to stipulate what the load case B might contain (remember it is still arbitrary). It is very tempting to suggest for load case B that it should represent a null load case, so that t*, u* and b* are all zero. This would indeed remove the last remaining volume integral, but would unfortunately remove the two surface integrals also, proving only that zero equals zero.

What turns out to be useful is to let load case B take the form of an infinitely concentrated point force at some position ‘p’ in the volume. This has three considerable advantages in helping us solve the integral equation above. These are:

1. For such a point force we know the displacement (u*) everywhere in the material. This is given by the ***fundamental solution*** developed by Lord Kelvin. The fundamental solution is sometimes termed the “free-space Green’s function”.

2. For such a point force, we also know the traction (t*) at all boundary points. This is obtained by differentiating the fundamental solution for u*.

3. For such a point force, the volume integral in equation (3) reduces mathematically to a single term, u(p), i.e. the displacement of the point p in the real load case A that we are solving.

The displacement fundamental solution for 2D linear elastic stress analysis is

$$u^*_{ij} = \frac{1}{8\pi G(1-\nu)}\left\{(3-4\nu)\ln\left[\frac{1}{r}\right]\delta_{ij} + \frac{\partial r}{\partial x_i}\frac{\partial r}{\partial x_j}\right\} \qquad (4)$$

where:

- subscripts i and j represent directions (i.e. x or y)
- u^*_{ij} is the displacement in direction 'i' at some location due to a concentrated point force in the direction 'j' at 'p'
- G is the material's shear modulus
- r is the distance from the point 'p' to the point at which the displacement is required
- δ_{ij} is the Kronecker delta, which takes the value zero if $i \neq j$, and 1 if $i = j$

Some key points about this fundamental solution:

- For any 'source' point 'p' and any 'field' point at which the displacement is required, the expression is easily calculated from the geometry of the two points and the material properties

- As the distance 'r' tends to zero, the natural logarithm term approaches infinity. This causes some numerical difficulties with elements that face each other across a narrow gap. Concept Analyst includes some features to take care of this effect. This is discussed in the Concept Analyst User Guide.

By this choice of load case B, then, equation (3) reduces to

$$u(p) + \int_S t^* u\, dS = \int_S u^* t\, dS \qquad (4)$$

The volume integrals have all been removed, and the only term that remains in the equation that relates to the inside of the material is the first one 'u(p)', that is the displacement at the point inside the object at which we applied the concentrated point force for the fictitious load case B.

The important point now is to recognise that although we have specified the nature of the load case B, that it is a point force, the ***location*** 'p' of the point force is still ***arbitrary***. So now we move the point 'p' ***to the boundary***.

$$c(p)u(p) + \int_S t^* u\, dS = \int_S u^* t\, dS \qquad (5)$$

This has the effect of introducing a multiplier, c(p), as in equation (5). This introduces little extra difficulty, because c(p) = 0.5 on any smooth boundary. To be more precise, c(p) takes the value of $\theta/2\pi$, where θ is the internal angle subtended at point 'p'.

Now we have an equation that contains only boundary terms, because 'p' is on the boundary and the two integrals are over only the boundary surface S.

Equation (5) is called the Boundary Integral Equation.

3.5 The boundary elements

On inspection of the fundamental solution given in equation (4), it is readily seen that the integration required by the Boundary Integral Equation is going to be very difficult if we are to do it analytically. In practice, for all but the simplest cases, the integration has to be done numerically using an approximate method.

Engineers will be familiar with classical numerical integration methods such as the Trapezoidal Rule and Simpson's Rule. It is most common for BEM (and FEM) codes to use an approach called Gauss-Legendre Quadrature, which provides simplicity and accuracy. Like the Trapezoidal Rule and Simpson's Rule, it helps if we divide the region over which we are integrating into small subregions – the finer the subdivision the greater the accuracy. These subdivisions are the boundary elements.

So, when the boundary is subdivided into these elements, we can write a subdivided form of the Boundary Integral Equation in which the integrals are expressed as the sum of the integrals over all the elements, giving

$$c(p)u(p) + \sum_{elem} \int_{S} t^{*} u dS = \sum_{elem} \int_{S} u^{*} t dS \tag{6}$$

Another feature of the use of elements is that they allow us a convenient way of formulating the problem in terms of a set of unknown values. Like finite elements, boundary elements have nodes that are often placed at the end and at mid-points of the elements. These can define both the geometry of the element and the displacement, traction and stress variation over the element. Concept Analyst's default element type is the quadratic element, and this element is described here. It is illustrated in figure 4.

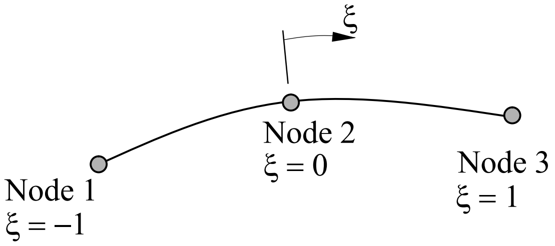

Figure 4. Quadratic boundary element

The three nodes all have (x,y) coordinates defining their location. These may be used to find the coordinates of any point on the element, and this is done using an interpolation procedure. For any value of the local coordinate ξ, as defined in figure 4, we can write an expression for the x-coordinate of the point at this location as

$$x = N_1x_1 + N_2x_2 + N_3x_3 \tag{7}$$

where the N_i terms are called shape functions (or interpolation functions) and are functions of the local coordinate ξ. For the quadratic element shown in figure 4, the shape functions may be given by

$$\begin{aligned} N_1 &= \tfrac{1}{2}\,\xi(\xi-1) \\ N_2 &= (\xi-1)(\xi+1) \\ N_3 &= \tfrac{1}{2}\,\xi(\xi+1) \end{aligned} \tag{8}$$

Clearly, the y-coordinate of any point on the element may be similarly interpolated.

3.6 The BEM as a matrix method

We have seen that the geometry of the element may be defined by interpolation from the coordinates of the nodes. An important step forward is to recognise that the displacement and traction distributions over the element may be likewise interpolated. Let us imagine, for the moment, that we know the x-direction displacement at each of the three nodes on the element, and denote these displacements u_1, u_2 and u_3. We would be able to find the displacement at any coordinate ξ using the interpolation

$$u(\xi) = N_1u_1 + N_2u_2 + N_3u_3 \tag{8}$$

or, expressing this as a vector multiplication,

$$u(\xi) = \mathbf{N}^T\mathbf{u} \tag{9}$$

where **N** is the vector containing the three shape functions and **u** is a vector containing the three nodal displacements. The superscript ‘T’ indicates the transpose of the vector.

It is next important to recognise that the expressions (8) and (9) still hold even if we do not know the nodal displacements. If these terms are variables, we can still express any displacement, for example, in terms of this interpolation between the variables. This is a very important step, because it allows the evaluation of the integrals in equation (6). At the moment, we cannot perform the integration because the integrals contain the terms u and t, both of which remain unknown. However, writing u and t in the interpolated form from equation (9)

$$c(p)u(p) + \sum_{\text{elem}}\int_S t*\mathbf{N}^T\mathbf{u}dS = \sum_{\text{elem}}\int_S u*\mathbf{N}^T\mathbf{t}dS \tag{10}$$

we can make the important step of taking the nodal displacement and nodal traction vectors, **u** and **t** respectively, out of the integrals. We can do this because the displacement at node 1, for example, takes a single value. As we proceed with the integration and traverse the element from $\xi = -1$ to $\xi = 1$, although the displacement at ξ will vary because of the shape functions, the nodal displacements do not. Removing the vectors **u** and **t** from the integrals, then, we can write

$$c(p)u(p) + \sum_{elem} \int_{S} t^* \mathbf{N}^{T} dS\, \mathbf{u} = \sum_{elem} \int_{S} u^* \mathbf{N}^{T} dS\, \mathbf{t} \qquad (11)$$

Remember that the point ‘p’ is some location on the boundary at which we place the concentrated point force in the fictitious load case B. Let us place this at a node, specifically node 1, and apply the load in the x-direction.

We know c(1) from the local geometry. We can calculate u* anywhere we like from the fundamental solution in equation (4). We can calculate t* anywhere we like from the traction fundamental solution, which for brevity is not stated in this document, but which is calculated by differentiating equation (4). For every element we can find values of the integrals, since every term in the integral is known. This leads to an expression as follows, in which it is assumed that c(p) takes the common value of ½ though this may be different if node 1 is at a corner.

$$\tfrac{1}{2} u_1 + h_{1,1}u_1 + h_{1,2}v_1 + h_{1,3}u_2 + h_{1,4}v_2 + \ldots = g_{1,1}t_{x1} + g_{1,2}t_{y1} + g_{1,3}t_{x2} + \ldots \qquad (12)$$

In other words, we have an equation relating the displacements u and v at each node to the tractions t_x and t_y at each node.

Obviously we cannot solve this equation because we have only one equation and many unknowns.

Now consider load case B to be a concentrated point force in the y-direction at node 1. Equation (11) is integrated again and this gives a different expression

$$\tfrac{1}{2} v_1 + h_{2,1}u_1 + h_{2,2}v_1 + h_{2,3}u_2 + h_{2,4}v_2 + \ldots = g_{2,1}t_{x1} + g_{2,2}t_{y1} + g_{2,3}t_{x2} + \ldots \qquad (13)$$

Place the point ‘p’ at every node in turn and apply the point force in both directions at each node, and a complete set of equations will be developed. These can be expressed in matrix form

$$\mathbf{Hu} = \mathbf{Gt} \qquad (14)$$

where **H** is a matrix of all the h coefficients, **G** is a matrix of all the g coefficients, and **u** and **t** contain the (as yet unknown) displacements and tractions at the nodes.

3.7 The solution

Equation (14) is a statement of a set of N simultaneous equations in which we have 2N unknowns. So the system still cannot be solved. At this point, we must reduce the number of unknowns, and this is done by applying a set of boundary conditions. The easiest way to do this is to recognise that each row of equation (14) contains two unknowns – a displacement and a traction – and we require the user to specify one or the other of these two variables.

In most cases this presents no major difficulty. A free surface will have a boundary condition of zero traction in both directions. At a displacement constraint, we typically know the displacement, but not the traction. Where a load is applied, we know the traction but not the displacement.

So applying our boundary conditions we arrive at a system shown graphically as follows:

$$\begin{bmatrix} h & h & h & h & h & h \\ h & h & h & h & h & h \\ h & h & h & h & h & h \\ h & h & h & h & h & h \\ h & h & h & h & h & h \\ h & h & h & h & h & h \end{bmatrix} \begin{Bmatrix} * \\ ? \\ ? \\ * \\ * \\ ? \end{Bmatrix} = \begin{bmatrix} g & g & g & g & g & g \\ g & g & g & g & g & g \\ g & g & g & g & g & g \\ g & g & g & g & g & g \\ g & g & g & g & g & g \\ g & g & g & g & g & g \end{bmatrix} \begin{Bmatrix} ? \\ \bullet \\ \bullet \\ ? \\ ? \\ \bullet \end{Bmatrix} \tag{15}$$

in which the h and g terms are known, the terms * are the displacements that have been prescribed as boundary conditions, the terms • are the tractions that have been prescribed as boundary conditions, and the terms ? remain unknown. The 'h' and 'g' terms in the square matrices are all denoted with the same letter in the graphical representation above (i.e. the subscripts from equations (12) and (13) have been dropped for clarity). It should be remembered, of course, that they all take different values since they represent the evaluation of different integrals.

We now swap columns of the matrices to bring all the displacements and tractions that remain unknown to the left hand side.

$$\begin{bmatrix} -g & h & h & -g & -g & h \\ -g & h & h & -g & -g & h \\ -g & h & h & -g & -g & h \\ -g & h & h & -g & -g & h \\ -g & h & h & -g & -g & h \\ -g & h & h & -g & -g & h \end{bmatrix} \begin{Bmatrix} ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{Bmatrix} = \begin{bmatrix} -h & g & g & -h & -h & g \\ -h & g & g & -h & -h & g \\ -h & g & g & -h & -h & g \\ -h & g & g & -h & -h & g \\ -h & g & g & -h & -h & g \\ -h & g & g & -h & -h & g \end{bmatrix} \begin{Bmatrix} * \\ \bullet \\ \bullet \\ * \\ * \\ \bullet \end{Bmatrix} \tag{16}$$

We now know all the terms on the right hand side of the equation, so we can multiply out the matrix vector product to leave

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{17}$$

where **A** is the square matrix on the left hand side of equation (16), containing columns of **H** and **G**, **x** is the vector of mixed displacements and tractions that remain unknown, and **b** is the vector result of multiplying out the right hand side of equation (16).

Equation (17) may be solved using a variety of techniques. Popular solvers for boundary element systems of equations are:

- ***Direct solver***. Usually Gauss elimination with partial pivoting. A combination of row operations reduces the system matrix to upper triangular form and then the solution is obtained through back-substitution. The advantage of this method is that it is guaranteed to arrive at a solution, though for large systems the computations should be done in double precision to avoid significant round-off errors.

- ***Iterative solver***. Usually the Generalised Minimum Residual Method (GMRES). This is a conjugate gradient type solver that is applicable to the nonsymmetric systems that characterise boundary element equations. The advantage of this method is speed, though occasionally the solver may fail to converge satisfactorily.

Concept Analyst uses a direct solver for the first analysis and the GMRES solver (with diagonal preconditioning) for most re-analysis runs.

The solution of equation (17) provides us with a complete description of the displacements and tractions around the boundary. The values of displacement over each element can be computed from the nodal values using equation (9), and the traction distribution may similarly be interpolated.

Some stress values may be derived directly from the tractions, for example a vertical element will have a direct stress σ_x given by the traction in the x-direction and a shear stress τ_{xy} given by the traction in the y-direction. For stress components that do not correspond directly to traction, the stress is determined from the strain components that may be found easily from the displacement variation over each element.

3.8 Internal points

The solution of equation (17) provides us with a complete description of the displacements and tractions around the boundary. However, Concept Analyst provides results throughout the material in the form of contour plots or internal line plots. In order to obtain results internally to the material to generate these plots, the program uses ***internal points***.

An internal point is simply a point inside the material, wholly contained within the closed loop of the boundary, at which we find results.

In order to explain how the internal point solutions work, let us return to the integral equation we developed immediately ***before*** the point 'p' was moved to the boundary. This was equation (4), which is restated here as equation (18).

$$u(p) + \int_S t^* u dS = \int_S u^* t dS \qquad (18)$$

This is a statement of the reciprocal theorem relating to:

- ***Load case A***: the real load case

- ***Load case B***: a concentrated point force at some ***internal*** location ‘p’

But once we have solved the system on the boundary, we know everything contained within the two boundary integrals. Evaluating the integrals, using the boundary elements again, we are left with a simple expression for the displacement, u(p), at the internal point.

Equation (18) can be differentiated to provide an expression for the stress components at the internal point.

3.9 Zones

A boundary element *zone* is a region surrounded by boundary elements. It has a set of material properties associated with it, and may have loads and constraints applied to it. Most *Concept Analyst* models consist of only one zone, which contains the entire boundary element model. However, there are occasions on which it may be desirable to define a two zone model:

- There are two different regions having different material properties
- There is a pin in a circular hole
- In some circumstances it may provide for faster run times

Consider a two zone problem as shown in figure 5. In this case the zones are adjacent regions sharing a straight line interface AB. The left zone, shaded grey (the default colour for mild steel) is bonded to an aluminium zone shaded in blue. The aluminium zone, zone 2, contains a circular hole. The conditions on line AB are:

- the displacements are continuous, so that the displacements on an interface element are the same in the steel as they are in the aluminium
- the tractions are equal and opposite, so that Newton’s 3rd law is enforced at each node on the interface.

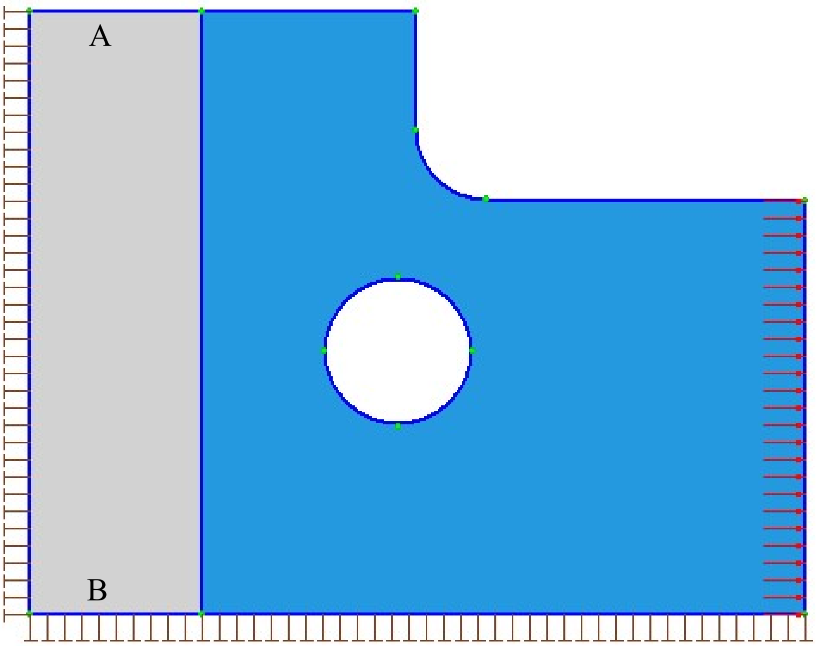

Figure 5. Two zone model example

This model is analysed by meshing as normal, though there is only a single line of elements/nodes on the interface, and these are shared between the two zones. Let us denote the displacements and tractions as follows:

u_1, t_1: displacements and tractions on the nodes wholly in zone 1
u_2, t_2: displacements and tractions on the nodes wholly in zone 2
u_{12}, t_{12}: zone 1 displacements and tractions on the nodes on the interface line AB
u_{21}, t_{21}: zone 2 displacements and tractions on the nodes on the interface line AB

Interface conditions are therefore:

$$u_{12} = u_{21} \tag{19}$$

$$t_{12} = -t_{21} \tag{20}$$

We build a set of boundary element influence matrices of the type shown in (14) and (15) for each zone. It is convenient to think of them in a partitioned form so it is clear which parts of the matrices multiply which terms in the vectors. For zone 1 we build the system

$$\begin{bmatrix} H_{1A} & H_{1B} \\ H_{1C} & H_{1D} \end{bmatrix} \begin{pmatrix} u_1 \\ u_{12} \end{pmatrix} = \begin{bmatrix} G_{1A} & G_{1B} \\ G_{1C} & G_{1D} \end{bmatrix} \begin{pmatrix} t_1 \\ t_{12} \end{pmatrix} \tag{21}$$

and for zone 2

$$\begin{bmatrix} H_{2A} & H_{2B} \\ H_{2C} & H_{2D} \end{bmatrix} \begin{pmatrix} u_{21} \\ u_2 \end{pmatrix} = \begin{bmatrix} G_{2A} & G_{2B} \\ G_{2C} & G_{2D} \end{bmatrix} \begin{pmatrix} t_{21} \\ t_2 \end{pmatrix} \tag{22}$$

These are combined to give a single matrix expression:

$$\begin{bmatrix} H_{1A} & H_{1B} & 0 \\ H_{1C} & H_{1D} & 0 \\ 0 & H_{2A} & H_{2B} \\ 0 & H_{2C} & H_{2D} \end{bmatrix} \begin{pmatrix} u_1 \\ u_{12} \\ u_2 \end{pmatrix} = \begin{bmatrix} G_{1A} & G_{1B} & 0 \\ G_{1C} & G_{1D} & 0 \\ 0 & -G_{2A} & G_{2B} \\ 0 & -G_{2C} & G_{2D} \end{bmatrix} \begin{pmatrix} t_1 \\ t_{12} \\ t_2 \end{pmatrix} \tag{23}$$

where the minus signs arise because of equation (20) being applied.

Now the t_{12} sub-vector is compressed to remove duplicates at shared nodes on the interface. It is assumed here that, while tractions on the boundary may be discontinuous because of step changes in the load applied, the internal tractions cannot become discontinuous. The tilde symbol '~' indicates the associated compression of a submatrix or vector.

$$\begin{bmatrix} H_{1A} & H_{1B} & 0 \\ H_{1C} & H_{1D} & 0 \\ 0 & H_{2A} & H_{2B} \\ 0 & H_{2C} & H_{2D} \end{bmatrix} \begin{pmatrix} u_1 \\ u_{12} \\ u_2 \end{pmatrix} = \begin{bmatrix} G_{1A} & \widetilde{G}_{1B} & 0 \\ G_{1C} & \widetilde{G}_{1D} & 0 \\ 0 & -\widetilde{G}_{2A} & G_{2B} \\ 0 & -\widetilde{G}_{2C} & G_{2D} \end{bmatrix} \begin{pmatrix} t_1 \\ \widetilde{t}_{12} \\ t_2 \end{pmatrix} \tag{24}$$

We now compress also the u_2 sub-vector to remove duplicate displacement degrees of freedom at the interface ends ('boundary interface nodes'). These displacements will of course be the same in zone 1 as they are in zone 2.

$$\begin{bmatrix} H_{1A} & H_{1B} & 0 \\ H_{1C} & H_{1D} & 0 \\ H_E & H_{2A} & \widetilde{H}_{2B} \\ H_F & H_{2C} & \widetilde{H}_{2D} \end{bmatrix} \begin{pmatrix} u_1 \\ u_{12} \\ \widetilde{u}_2 \end{pmatrix} = \begin{bmatrix} G_{1A} & \widetilde{G}_{1B} & 0 \\ G_{1C} & \widetilde{G}_{1D} & 0 \\ 0 & -\widetilde{G}_{2A} & G_{2B} \\ 0 & -\widetilde{G}_{2C} & G_{2D} \end{bmatrix} \begin{pmatrix} t_1 \\ \widetilde{t}_{12} \\ t_2 \end{pmatrix} \tag{25}$$

The submatrices H_E and H_F are mostly comprised of zeros, but include non-zero columns that multiply the boundary interface node displacements. Now move the subvector t_{12} and associated ***G*** submatrices to the left hand side

$$\begin{bmatrix} H_{1A} & -\widetilde{G}_{1B} & H_{1B} & 0 \\ H_{1C} & -\widetilde{G}_{1D} & H_{1D} & 0 \\ H_E & \widetilde{G}_{2A} & H_{2A} & \widetilde{H}_{2B} \\ H_F & \widetilde{G}_{2C} & H_{2C} & \widetilde{H}_{2D} \end{bmatrix} \begin{pmatrix} u_1 \\ \widetilde{t}_{12} \\ u_{12} \\ \widetilde{u}_2 \end{pmatrix} = \begin{bmatrix} G_{1A} & 0 \\ G_{1C} & 0 \\ 0 & G_{2B} \\ 0 & G_{2D} \end{bmatrix} \begin{pmatrix} t_1 \\ t_2 \end{pmatrix} \tag{26}$$

This is manipulated in a similar way to that described in section 3.7, i.e. boundary conditions are applied to reduce the number of unknowns to equal the number of equations, and this combined system is solved to give the displacements and tractions that were not specified as boundary conditions.

3.10 Self weight

If the analysis is to consider self-weight loads, or loads from any given constant acceleration experienced by the body under analysis, it is necessary to add an extra volume integral term to the boundary integral equation, i.e.

$$c(p)u(p)+\int_S t*udS=\int_S u*tdS+\int_V u*bdV \tag{27}$$

in which b* represents a general body force term, so that in the case of self-weight it will comprise the effects of the distributed mass in the body undergoing acceleration. Notice that the integration must take place over the volume of the object, so that some further action will be necessary to implement this is a boundary-only framework such as Concept Analyst. The most effective way to do this is through the "Galerkin vector" approach, which transforms the volume integral mathematically into an equivalent surface integral. This results in (27) being modified to

$$c(p)u(p)+\int_S t*udS=\int_S u*tdS+\int_S PdS \tag{28}$$

in which P is a function of the geometry and material properties, rather like the fundamental solution (4). The integral of P can easily be evaluated to give a term on the right hand side to be added to the boundary element system. This is carried through the rest of the theoretical development to give additional terms on the right hand side of the equation (17) that is solved to give the stresses and displacements.

It is important to recognise that if the above approach is used to compute the effects of body forces, the fundamental solution for displacement must include an extra constant term, i.e.

$$u_{ij}^{*}=\frac{1}{8\pi G(1-\nu)}\left\{(3-4\nu)\ln\left[\frac{1}{r}\right]\delta_{ij}+\frac{\partial r}{\partial x_i}\frac{\partial r}{\partial x_j}+\left(\frac{7-8\nu}{2}\right)\delta_{ij}\right\} \tag{29}$$

3.10 Crack models

The conventional method of collocating equation (11) at all nodes will fail for crack models since there are nodes on opposing crack surfaces that are in the identical location. In this case, then, there will be duplicate equations in the set of simultaneous equations derived using the BEM, so we have fewer independent equations than unknowns and no solution is possible. If the model contains cracks, then, Concept Analyst uses a variant of the BEM known as the Dual Boundary Element Method (or Dual BEM).

Here the displacement boundary integral equation (11) is used when collocating at all nodes on one crack surface (and when collocating at all non-crack nodes), and the traction boundary integral equation is used when collocating on the opposing crack surface. The traction boundary integral equation is derived by differentiating the displacement boundary integral equation. This provides for different equations when collocating at coincident nodes on opposing crack surfaces and allows a solution to be obtained.

The use of the traction integral equation means that the boundary integrals are of a higher order of singularity than in the conventional equation (11), and this has implications on the analysis. In particular, the equation is valid only if there are certain continuity requirements satisfied at the collocation point. These requirements are observed by the use of discontinuous boundary elements, in which the nodes are no longer at the element ends (and therefore shared by adjacent elements) but are now moved away from the element ends, so that they are not shared by adjacent elements.

The mode I and mode II stress intensity factors, K_{I} and K_{II} respectively, are calculated using the J-integral approach. This involves evaluating the integral of a function of strain energy over a path surrounding the crack tip. This is done over a circular path in Concept Analyst, the path centred on the crack tip and the radius of the circle being given by the distance from the crack tip to the mid-point of the second element on the crack surface (i.e. the element adjacent to the element touching the crack tip). Notice that as the crack tip approaches the boundary of the object, this may fail because the path leaves the problem domain.

3.11 Crack growth

Crack growth is assumed to be through the fatigue mechanism of sub-critical growth under oscillatory loading. Cracks can propagate until they become critical, i.e. when the stress intensity factor, K_I, reaches the fracture toughness, or when they break through another boundary. Breaking through is deemed to occur when the size of the plastic zone that develops ahead of a crack tip exceeds the distance to that boundary. Although Concept Analyst is only performing linear elastic analyses, the plastic zone size can still be estimated.

There are many crack growth laws that have been determined empirically, but Concept Analyst uses the Paris law, which states that the crack growth rate per cycle, da/dN, is given by $C(\Delta K)^m$, where C and m are material constants, and ΔK is the range of stress intensity factors experienced over the loading cycle. If the loading cycle consists of both tensile and compressive loads, the compressive part is neglected in calculating ΔK since the crack may be assumed to be closed and no longer acting as a crack.

ΔK is computed in Concept Analyst by making use of the stress ratio, R, which is the minimum stress applied in the loading cycle divided by the maximum stress in the cycle.

At each crack increment, cracks are assumed to propagate in the direction perpendicular to the maximum circumferential stress, and a predictor-corrector algorithm applied to give the correct direction independent of the crack growth increment. For this reason, several analysis runs can be performed within each crack increment before proceeding to the next increment.

4

Implementation in Concept Analyst

4.1 Introduction

Concept Analyst is a Windows based software system that uses the Boundary Element Method to solve two-dimensional problems in elastostatics. It is written in the language C++ and makes use of the Microsoft Developer Studio foundation classes.

Unlike most analysis systems, it does not require the user to build the geometry by defining first points, then lines, then elements, etc. Instead, it uses higher level geometric entities called shapes; these may be circles or rectangles, and rectangular shapes may then be modified by the addition of more points to generate irregular closed polygons. Any vertex or vertices of these polygons may be given a fillet radius, or otherwise modified to become a circular arc. Further, any edge of a general polygon may be modified to be defined as a B-spline. Circular shapes may be modified only by redefining their diameter.

Decisions about meshing are taken by the program, which contains a set of automatic meshing routines that have been carefully developed to produce appropriate meshes for a wide range of geometries. All elements are quadratic, 'isoparametric' line elements. The user may also select a coarse, standard or fine mesh, thereby reflecting the balance sought between the run time and the accuracy.

In general, the use of the default setting, i.e. standard mesh, should give a good engineering balance.

4.2 Element type and shape numbering

The quadratic element used by Concept Analyst is shown in figure 5.

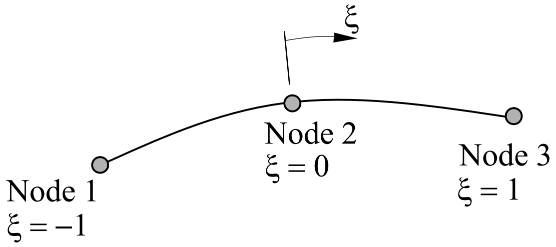

Figure 5. Quadratic boundary element

The shape functions for the quadratic element are:

$$N_1 = \tfrac{1}{2}(1 - \xi)$$
$$N_2 = (1 - \xi)(1 + \xi)$$
$$N_3 = \tfrac{1}{2}(1 + \xi)$$

The element is oriented in Concept Analyst such that in travelling from node 1 to node 3, the material to be analysed is on the left. This means that elements are defined in a counter-clockwise direction around the outer boundary and in a clockwise direction around holes and any other internal boundaries. This direction of definition is generated automatically by Concept Analyst. The outer shape, which is also determined automatically, is then defined in a counter-clockwise direction, and all others in a clockwise direction.

4.3 Automatic meshing

Concept Analyst defines elements according to the following scheme, which is based on keeping track of the length of the element at each end of every line.

1. Set the maximum element size to be equal to 2.5%, 5%, 9% or 17% of the largest problem dimension for (respectively) superfine, fine, standard and coarse mesh density. Apply this to both ends of all lines.
2. Identify re-entry corners and set element size in those areas to 10% or 20% of the maximum element size, depending on the severity of the angle.

3. Identify short line segments and arcs. Set an appropriate element length at each end to ensure that a minimum number of elements is defined on the line. The minimum number is 1 for a straight line and varies for circular arcs depending on the user-selected mesh density.
4. Compare element sizes at the two lines that meet at each vertex. Reduce the large to 1.5 times the smaller if it is greater than this value.
5. Compare start and end elements on each line. If elements cannot be graded along the length of the line such that no element is no more than 1.5 times the length of its neighbour, reduce the larger end element size to accommodate this limit on grading.
6. If any changes have been made in steps 4 and 5, return to step 4.
7. Generate the mesh for the model from the element sizes at the end of each line.
8. Loop through all elements to ensure no element is too long in comparison with the distance from a node on another element. If such a case is found, the element concerned is subdivided, and may be subdivided again, until each element is no longer than twice the distance to the nearest node.

4.4 Internal point definition

Concept Analyst defines internal points automatically over the region enclosed by the boundary. As shown in section 3.8, these are points at which the program computes the displacements and stresses after the boundary solution has been found. This involves a small amount of extra computation for most models. The internal points are used for one major reason – that they provide information on the stress and displacement results inside the material so that accurate contour plots can be drawn. For the plots to be reasonably accurate:

- There should be a sufficient number of internal points to cover the area of the material
- There should be a concentration of internal points in regions of high stress gradient, e.g. around fillets, holes and other stress risers.
- The internal points should not be placed too close to the boundary (within about one quarter of the element length).

The program defines internal points automatically when the Analyse command (‘Go’ button) is selected. This is done using the following scheme:

1. Define a ring of internal points around every fillet
2. Define four concentric rings of internal points around each circular hole
3. Define lines of internal points along lines that are likely to act as a neutral axis in bending (this gives improved contour plots for problems dominated by bending).
4. Scatter more internal points at random over the material, using a total number of internal points the same as the number of nodes (this has been found to give a reasonable number of points for most problems)
5. Triangulate to generate a triangular mesh joining all nodes and internal points.

6. Smooth the internal points iteratively so that the mesh of triangles is smooth and well graded

Internal points are not displayed on the screen. It is not possible to set up the program so that internal points are not created.

The 'Report Quality Contours' feature, available in the 'Edit – preferences' menu, allows the user to generate a plot having improved clarity of contour definition. This is done by defining, in Step 4 (above), a total number of internal points equal to the maximum allowable by the program. The greater number of internal points gives greater coverage of the material region and higher definition contours.

If initiated when results are available and contours are being shown, this feature causes a re-analysis to be performed so that the extra run time is short. Otherwise the greater number of internal points will be used for the first analysis after the feature is set.

The 'Report Quality Contours' feature is automatically turned off on the subsequent full analysis or re-analysis. It is also not available to be set as the user's default setting. This is because the standard quality contours are sufficient for most purposes, and run considerably more quickly. It is intended, therefore, that users obtain the contour plots they require and then, just before writing contour plots to the report file, set the report quality contours if greater contour definition is required. However, once an analysis has been performed using report quality contours, the high definition plots may be displayed for multiple contour plots of different quantities until a new analysis or re-analysis is performed.

4.5 The analysis

On selection of the 'Analyse' command ('Go' button) the analysis proceeds according to the theory as defined in section 3 of this guide. The mesh is produced, and also internal points, and integration begins in formation of the governing matrix.

Almost all of Concept Analyst's calculations are performed in core memory so that performance is optimised. Notice also that the architecture of the program is such that the graphical and analysis features of the software are contained in the same program and share the same database. This precludes the need for writing and reading of data and results files, which for small analysis problems comprise a significant portion of the run time.

The matrices **H** and **G** are not normally formulated explicitly; instead, the matrix **A** and vector **b** are formed during the integration phase by considering the boundary condition on the field element during each source point – field element integration. This saves a considerable amount of memory space and run time. The **A** matrix is stored to a file once it is generated completely, and before the equation solution starts.

4.6 Adaptive mesh refinement

Concept Analyst includes an option for the program to undertake adaptive mesh refinement. This is an iterative process that uses multiple analysis runs of the same problem, with progressively improving meshes, in order to converge to an improved set of results.

There are two main components of an adaptive analysis scheme:

> ***Error indicator***. It is important to estimate the level of accuracy in a boundary element solution. This is used as both a stopping criterion, when the error has diminished to a sufficiently small value, and a guide towards mesh refinement if significant errors still remain. Thus it is useful to have both global and local properties – a global error estimator is useful as a stopping criterion and a local error estimator is required if we are to refine the model efficiently.
>
> ***Mesh refinement scheme***. If the error indicator shows that improved results would result from refinement of the mesh in a certain area of the model, there are various different approaches to carry out that refinement. Classical '*h*-refinement' involves subdivision of existing elements so that more elements are used where they are required. This is the approach used in Concept Analyst. Another approach called '*p*-refinement' involves improving the model by increasing the order of the boundary elements used, e.g. if the errors exist on a linear element, then in the next iteration it might be made quadratic.

The error indicator used in Concept Analyst is based on the von Mises stress, which is an expression used in the von Mises failure criterion (section 2.4.3). The von Mises stress, σ_{VM}, is given by

$$\sigma_{\mathrm{VM}} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}$$

so it is based on the square root of the sum of the squares of the three Mohr's circle diameters. Von Mises stress is often used as a useful quantity to display because it gives a single stress value at a point, that relates directly to a widely used failure criterion, and which remains positive for both tensile and compressive stress states.

The error indicator is based on the discontinuity in von Mises stress between neighbouring elements. The stress can be computed over each element totally independently, and this will result in two values being available at the end of each element. In graphical displays of results, these two values are averaged to give a smooth display. However, it has been found that the size of the discontinuity provides a useful error estimator.

In the adaptive option, an element will be subdivided if:

- The discontinuity in von Mises stress at one or both of its ends is greater than 1.5% of the maximum von Mises stress anywhere in the model, and

- The range of von Mises stress over the element is greater than 1.5% of the maximum von Mises stress anywhere in the model, and

- The subdivision would not give rise to an element shorter than 1.25% of the length of the line on which it lies.

An element *i* would also be subdivided if:

- An adjacent element $i-1$ or $i+1$ becomes subdivided to a length less than one quarter of the length of element *i*.

If, in a particular iteration, no elements satisfy the above requirements, the program deems the solution satisfactory.

The percentages used in the above criteria have been determined through a series of numerical tests.

An adaptive analysis may use as a starting point the Coarse, Standard or Fine mesh setting (selected in Edit – Preferences…). Use of a coarse mesh is recommended since the adaptive refinement will proceed rapidly and will make a more optimal use of the elements in the final mesh.

4.7 Re-analysis

Once the analysis has been completed and the contours of results displayed on the screen, many types of geometric change to the design will then cause a re-analysis to be initiated. This means that a very rapid analysis is performed, in which as much as possible of the previous analysis computation is re-used so that updated results are presented as soon as possible.

On most geometric changes in 2D, the changes are limited to a comparatively small portion of the boundary. This means that the boundary element mesh for the new model is to a large extent identical to that used for the previous model. For example, figure 6 shows a change in fillet radius. The elements on the fillet itself, and those on the two adjacent tangent lines, are the only elements modified.

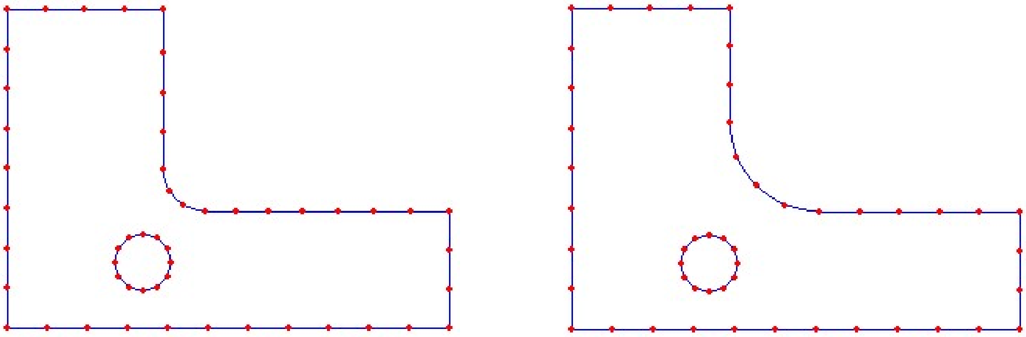

Figure 6. Modification of fillet radius and consequential mesh changes

As a result of this, the **A** matrix will also be largely similar to that generated in the previous analysis. Only the shaded portion of the **A** matrix (shown in figure 7) is required to be updated (these are the rows and columns relating to the modified elements and nodes).

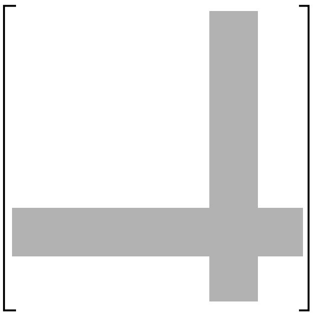

Figure 7. Modified portion of **A** matrix shown shaded

This clearly offers some considerable time saving over the full analysis. The solution phase then follows using an iterative solver (diagonally preconditioned GMRES). In the iteration, the first guess at the solution is provided by the previous solution. Since this is often similar to the revised solution to the updated model, the iteration is offered the benefit of a good start.

Figure 8 shows the time savings to be gained by using re-analysis over a full analysis. The saving will depend strongly on the extent to which the analysis model is changed, i.e. the proportion of the total number of elements that are moved. The figure shows results from a mixed set of problems of different sizes and different character, and shows the proportion of the full analysis time that is required to complete the re-analysis.

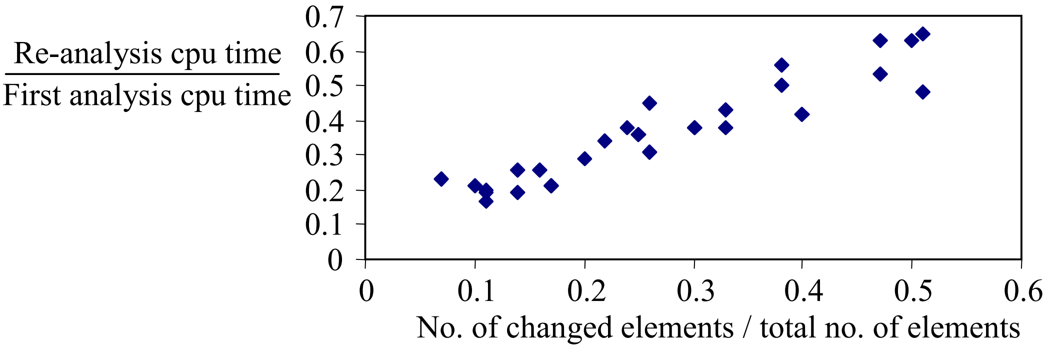

Figure 8. Time savings to be achieved by re-analysis

4.8 Acceleration of re-analysis for dynamic contour display

Version 1.9 and later of the software allows the facility for contour displays of results to be updated rapidly so that for small problems the display is updated in real-time as a geometric change is being carried out. For example, as a hole is being moved the maximum principal stress contours may be continuously updating to reflect the results for the latest geometry. This is the result of a recent research programme into acceleration of boundary element computations.

The three major numerically intensive stages of a boundary element analysis are

- Integration of the boundary integrals to find terms that will populate the system matrix **A**.
- Solution of the set of simultaneous equations in matrix form, **Ax** = **b**.
- The solution at internal points, which again involves evaluation of boundary integrals.

All three phases have been accelerated, though the acceleration of the first and third phases are identical in method as they relate to integration.

Integration has been accelerated by the use of surface fit expressions to approximate the values of the boundary integrals in (11). These integrals are functions of:

- Material properties
- The location of the source point ‘p’
- The local geometry and orientation of the element over which the integration is taking place

If we keep the material properties constant, e.g. take the elastic constants for mild steel, we can consider the integrals to be functions of a few geometric variables, i.e.

- Angle θ, being the anticlockwise angle from the horizontal of the vector from the source point ‘p’ to the mid point of the element
- Angle ϕ, being the orientation of the element
- Non-dimensional distance R_m, being the distance from point ‘p’ to the mid-point of the element divided by the element length.

The values of the boundary integrals vary smoothly as these geometric parameters are changed, and can be displayed as a surface plot on axes of these geometric variables. Using the method of least-squares, a regression fit can be made to these surfaces. Using care to select the most appropriate functions to include in the fit, some extremely efficient expressions can be found. For example, the integral for the mid-node of a vertical element may be expressed as

$$U_{001} = \left(2.166 + 2.158\cos 2\theta - 8.996\ln R_m\right)\times 10^{-7}$$

It is considerably faster to evaluate this expression than to use the conventional numerical integration procedures. A total of 384 surface fit expressions have been generated, using the elastic properties of mild steel, so that all integrals over flat elements may be considered in this way in Concept Analyst.

The other numerically intensive phase of the computation, the solution of the system of equations, has been accelerated by use of an iterative solver. The GMRES solver is used, as in previous versions of Concept Analyst for re-analysis, but from version 1.9 a change has been made to the way the solver is ‘preconditioned’. Preconditioning is modification of the system of equations being solved to make them more amenable to iterative solution, and thereby improving the convergence rate of the iteration to solution.

The best preconditioner that can be found to solve the system **Ax** = **b** is the inverse of **A**, i.e. $\mathbf{A}^{-1}$. This will guarantee convergence in only one iteration. However, we do not have this inverse (if we had, we would simply use it to solve the equations directly using $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$). What is needed is an approximation to the inverse of **A** that may be found quickly and easily.

The strategy used in Concept Analyst is to solve the full analysis (on pressing Go) by use of an LU-decomposition method. This produces a lower triangular matrix, **L**, and an upper triangular matrix, **U**, that multiply together to give **A**. It is straightforward to find the decomposition. Since the matrices are triangular the solution to $\mathbf{LUx} = \mathbf{b}$ can be found by two backsubstitution phases as in the final stage of a Gauss elimination solver. These matrices **L** and **U** are stored and are later used as a very effective preconditioner for solving the re-analysis system in which a modified matrix **A** is to be solved. Typically the converged solution to the re-analysis problem is found in a few iterations, giving a re-solve time 10-30% of the time it would take to solve the system using, say, Gauss elimination.

4.9 Theoretical limitations

At this point we present one fundamental problem with boundary elements; the problem of elements close together back-to-back. The problem results from the fact that it is necessary to perform numerical integrations of the fundamental solutions over elements.

The form of the fundamental solution for two-dimensional stress analysis is given in equation (4). Notice that it contains a term ln(1/r), where ‘r’ is the distance from the node ‘x’ at which we place the fictitious point force to the point ‘y’ at which we evaluate its effect. Other fundamental solutions for other types of problem also exhibit similar behaviour, in that the denominator contains ‘r’, or sometimes r^2 or even r^3.

Consider now the situation in which two elements face each other back-to-back across a narrow gap, as shown in figure 9. The material may be outside the elements, as in the case of a narrow slot, or in between the elements, as in the example of a narrow strip of material. It does not matter; the problem remains.

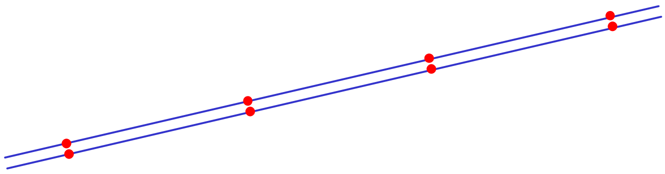

Figure 9. Elements close together

If the point force is at ‘x’ on one of these elements, and we are integrating over the other element, then the distance ‘r’ is going to approach zero somewhere over the range of integration. As $r \rightarrow 0$, then the fundamental solution will tend to infinity. It causes problems for most integration schemes when the function being integrated passes close to infinity.

For this reason, users will find that Concept Analyst automatically defines more elements on thin sections than it will in cases with a lower surface area/volume ratio. The reason is that the important factor is (element size ÷ separation). In practice, it is desirable for the element size to be no more than around twice the separation. Problems containing more elements will require more run-time to solve.

For related reasons, boundary elements are not well suited to the analysis of very thin sections. It is advised that the aspect ratio of objects being analysed, or of parts of objects being analysed, should not exceed 20.

5

Comparison with the Finite Element Method

5.1 Introduction

As has been described throughout this document, Concept Analyst calculates the stress and displacement fields for general problems using the Boundary Element Method (BEM). However, this type of calculation is most commonly performed using the Finite Element Method (FEM). This section compares the two methods.

5.2 The Finite Element Method

In brief, the FEM is similar in concept to the BEM, but instead of defining only the surface area of the object, the entire volume must be defined. This means that 3D models must be defined in terms of three-dimensional brick elements, and that the 2D problems analysed by Concept Analyst would require two-dimensional finite elements (triangles and quadrilaterals) to be fully defined.

The method is, like the BEM, based on the use of matrix algebra to solve large systems of simultaneous equations. It also uses the concept of node points to define the displacement on each element, and the use of shape functions to describe the variation of this displacement over the element. These shape functions are identical to boundary element shape functions, though of course they have to consider an extra dimension.

The FEM relies on the idea of stiffness, k, being defined as a fundamental relationship between force, f, and displacement, u.

$$f = ku \tag{30}$$

Equation (30) might easily be applied to a one-dimensional problem such as a spring in longitudinal extension, so that the extension of the spring may easily be found for any value of applied force.

For a two-dimensional, rectangular, finite element, we can derive an analogous expression

$$\mathbf{f} = \mathbf{ku} \tag{31}$$

in which **f** is now a vector of forces applied at the nodes on the element and **u** is a vector containing the nodal displacements. Both the force and displacement vectors need to be compiled containing components in the x- and y-directions. The term **k** represents a square 'stiffness matrix' defining the behaviour of the element, and this will be a function of the element geometry and material properties.

In practice, we determine the stiffness matrix, **k**, for each element by performing an integral over the element, and (like BEM) this is most commonly done using Gauss quadrature.

Of course, nodes will be shared between adjacent elements, and therefore the displacement of any one node will be represented in more than one of the element-based equations of the form of equation (31). In order to solve the problem so that all element stiffness equations are satisfied, it is necessary to build them into a single, 'global' system

$$\mathbf{F} = \mathbf{KU} \tag{32}$$

where **F** is a vector containing the forces at every node in the model, **U** is a vector containing the displacements at every node in the model, and **K** is a square stiffness matrix containing all the terms in the individual element stiffness matrices. For large problems, then, the matrix **K** can become very large indeed. Fortunately, if the problem is carefully defined so that no element contains two nodes with very different node numbering, the matrix becomes strongly 'banded', as shown in figure 10.

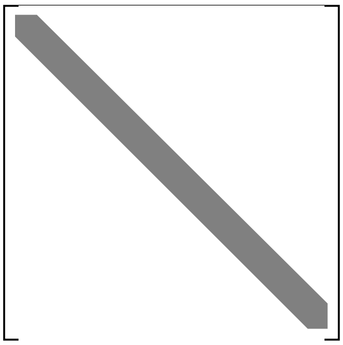

Figure 10. Typical FEM stiffness matrix **K** showing banding

The non-zero terms are restricted to the area of the matrix shown shaded in the figure. This feature of the matrix makes it efficient in terms of both storage (since the many zero terms do not need to be stored in memory) and solution time (since there are a number of highly developed solution procedures for this type of matrix).

Once the system of equations (32) has been formed it needs to be solved to find the unknowns, which (unlike the BEM theory presented in section 3) will comprise only displacements. In order to do this, firstly the equations that correspond to displacement constraints should be eliminated from the system, otherwise the system will be singular and insoluble. The solution proceeds to compute the displacements at each node and in each direction.

From the displacements the strain components in each element may be found by differentiating displacement components with respect to the coordinate directions. From these strains it is straightforward to find the stress components using the principles of elasticity given in section 2.

5.3 Accuracy comparison

It is difficult to compare the accuracy of the BEM and FEM directly. Both methods require a sufficient number of elements to ensure the best accuracy, and since they are different techniques they will require different meshes in order to achieve any given accuracy. The number of elements required is strongly dependent on the problem being solved: the geometry, the nature of any loading, the presence of any discontinuities, the severity of any stress concentrations, etc.

What can be stated with some confidence is that for general problems, a boundary element model will contain fewer elements than a finite element model to achieve the same accuracy in stress results. In many cases, the number of elements will be far fewer. Of course, this is largely because the boundary elements are defined only on the surface, but still the boundary elements would normally be larger in size than the finite element size required for the same accuracy.

Since the boundary element method requires the user to define far less information (geometrically) than does the finite element method, it is perhaps surprising that the BEM is capable of producing solutions of great accuracy. An intuitive way of expressing this is to say: every element we use to define a model is an approximation to reality, and in the FEM we make these approximations throughout the volume, whereas in the BEM we make these approximations only on the surface. Fewer approximations, fewer errors.

In a little more analytical way, we can see some more theoretically based reasons for the extra accuracy in stress solutions that boundary elements can offer. In particular, this is evident from the way the stresses are calculated.

- Most FEM programs calculate first the displacements, then differentiate to obtain strains, then use elasticity relations to obtain the stresses. The differentiation is an important step, since the procedure reduces the order of the approximation. For example, a finite element model that is quadratic in displacement will be capable of giving only linearly varying strains, and therefore stresses.

- The BEM solves for the unknown vector **x** (from equation (17)), and this contains both displacement and traction components. Many stresses can be obtained directly from the tractions, and this is done without differentiation. (Some stress components still require differentiation).

Another reason for good stress accuracy from boundary elements is that the peak stresses almost always occur on the boundary in a linear elastic analysis. Since this is where the BEM calculates its solutions, it is natural to expect the accuracy to be at its best here. The most accurate results obtained in a finite element are inside its volume, and some approximations are involved in extrapolating these more accurate results to the boundary.

5.4 Versatility

Without question, the single most important drawback to boundary element methods is their range of applications. Certainly in the commercial BEM software available today, there is very limited capability to solve problems involving non-linear material properties or time dependence.

Here the FEM has a great advantage, being able to solve a much wider variety of problem types.

5.5 Usage of computer resources

Since the BEM uses far fewer elements, and therefore has far fewer degrees of freedom than the corresponding finite element model, it is natural to expect it to require far less memory, disc storage and computation time. This is true to some extent, but not completely true.

The memory and disc space requirements are less than for the FEM for most problems of the type that Concept Analyst is typically used for. Although the BEM matrix is full and unsymmetric, and therefore needs to be stored in its entirety, it is considerably smaller than the corresponding finite element global stiffness matrix, so that the storage requirements may be reasonably be expressed as the shaded areas in the matrices shown in figure 11.

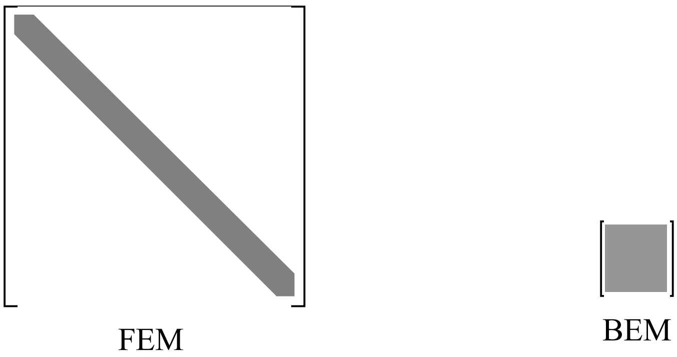

Figure 11. Storage requirement comparison

As the problem size becomes larger, especially in three dimensions, the fact that the BEM matrix in full, and not banded, becomes increasingly important and the disc storage requirement may become excessive.

Users should not expect BEM run times to be much less than those for the FEM. Often the solution times are approximately the same. Sometimes the BEM can require considerably longer than the corresponding FEM model.

5.6 Automation and re-analysis

Concept Analyst contains a high degree of automation of the decisions that have traditionally been made by expert analysts, including the type of elements to use, the number of elements and the way these elements are distributed around the geometry in order to solve most accurately and efficiently for stress concentrations. As has been seen in this document, the program also makes extensive use of re-analysis to update stress results as a design geometry evolves.

It is clearly simpler to place boundary elements on a geometry more reliably and robustly than it is with finite elements. This is important in a system that relies so heavily on reliability of automatic meshing. Furthermore, when a re-analysis is initiated after a geometric modification, it is straightforward (at least in a 2D problem) to revise the boundary element mesh to accommodate the change. It is principally for these reasons that the BEM was adopted as the analysis tool of choice for Concept Analyst.

It should be noted that Concept Analyst does create an internal ‘mesh’ through the area of the material, in a similar way to finite elements. The vertices of this mesh are the locations of internal points, and the triangular mesh is used for the display of contours of results. At first sight, this might imply that it might be just as efficient to define a finite element mesh. To an extent this is true. However, the triangular mesh joining the internal points does not have to conform to finite element meshing rules. The triangles can be elongated or otherwise distorted without detracting from the results. Thus, although an internal ‘mesh’ is created for each analysis by Concept Analyst, there are still powerful reasons why boundary elements might be preferable for this application.