A thesis submitted to the University of Cape Town in partial fulfilment of the requirements for the degree of Master of Science in Engineering.

A NEW METHOD OF MESHING IN DISCONTINUOUS DEFORMATION ANALYSIS (DDA)

by

D E Clatworthy

B.Sc. Eng (Civil) (Hons)
UCT, 1993

Department of Civil Engineering,
in association with the Centre for
Research in Computational and
Applied Mechanics (CERECAM);

University of Cape Town,
December 1997.

Declaration

I, David Clatworthy, hereby declare that this thesis is my own work and that it has not been submitted for a degree at any other university.

Signed by candidate

D E Clatworthy
December 1997

Acknowledgements

The bulk of the funding for my postgraduate studies came from the Chris van Breda Scholarship, administered by the Postgraduate Scholarships Office of the University of Cape Town. Ninham Shand (Pty) Ltd set up this scholarship in association with the parents of Chris van Breda, a young civil engineer who was killed in a motor vehicle accident in 1982.

I received additional funding from the Foundation for Research and Development (FRD) through the Centre for Research in Computational and Applied Mechanics (CERECAM).

To these people and organisations I am very grateful.

My supervisor, Dr F Scheele, was an invaluable source of advice, support and encouragement throughout my studies.

I would also like to thank my 'office-mates', Johnny Lai Sang and Charmaine Rose for their help and company over the last two years. Chantal Walti gave me much-needed advice and help in putting together this report, and Allison Zelkowitz provided much of the inspiration.

I am deeply grateful to all five of these people for their friendship during my time in Cape Town, and beyond.

Finally, I would like to express my profound appreciation for all the support provided by my parents and family over the past years.

Synopsis

Discontinuous Deformation Analysis (DDA) is a discrete element method developed by Shi [1988] specifically for modelling blocky rock masses. The DDA method is based on the assumption that deformation and failure of such rock masses is primarily due to differential movements of rock blocks, rather than strain and fracture of intact rock material. Strains and stresses are assumed to be constant over the area of each rock block. Contact between blocks is modelled using penalty functions, with Coulomb's friction law controlling sliding along block boundaries.

Tests show that while DDA is not well suited to dynamic simulations where the velocities of blocks become large, it can model rock masses to a reasonable degree of accuracy in static analyses. There are various analysis control parameters which have a marked effect on the solution, however, and the user should take care in choosing suitable values for these parameters.

A method is proposed here, in which certain blocks can be sub-divided into Finite Element meshes in order to obtain a more accurate description of their deformation. The method takes advantage of the fact that both DDA and the Finite Element Method (FEM) use the principle of stationary potential energy to obtain the solution equations for block equilibrium. Both DDA blocks and FEM elements can therefore initially be treated as DDA blocks, using the standard DDA formulation, and then the solution equations for the FEM elements are converted into Finite Element format by a simple transformation procedure before solution.

First and second order DDA blocks are considered in this report, along with their equivalents in FEM, the C^0-linear and C^0-quadratic triangular elements. The C^0-linear elements are found to be too stiff in modelling bending deformation, due to the assumption of constant strain throughout the element. The C^0-quadratic elements are able to accurately model bending, however. It is shown through tests that the performance of these FEM elements, formulated within the DDA method, is identical to that obtained using the corresponding elements in conventional Finite Element programs.

The sub-meshing method therefore allows mixed-formulation analyses, with DDA blocks and FEM meshes interacting within a single system, while remaining efficient, and reasonably simple to incorporate into existing DDA program codes. It would also be possible to model material non-linearity and fracture using this method.

Table of Contents

List of Figures

List of Tables

Glossary of Symbols

Throughout this report, two-dimensional matrices are represented by a capital letter, with double underlining. One-dimensional column arrays are written as a lower-case letter, underlined once only. Where an array is specific to a particular block, the block number is referred to by a superscript. A single term in an array is written with subscript indices.

Array sizes refer to 1st order blocks and elements, and may be different for the 2nd order formulation.

Standard SI units are listed here, using the assumption of unit depth in the z-direction. However, any consistent system may be used.

Symbol	Description	Units
$\underline{\underline{B}}^i(x, y)$	Strain-displacement matrix for point (x, y) (3 x 12 array).	
C	Cohesion of discontinuity.	Pa
$\underline{d}$	Deformation vector in solution equations ($6n$ x 1 array).	
$\underline{d}^i$	Deformation vector for block i (6 x 1 array).	
d_r^i	r-th term in deformation vector of block i.	
d_p	Penetration distance at block contact.	m
d_s	Sliding distance at block contact.	m
$\underline{\underline{D}}$	Material stiffness array, or tangent modulus (3 x 3 array).	Pa
$\underline{e}$	Displacement vector for contacts and bolts (6 x 1 array).	
$\underline{\underline{E}}$	Material stiffness array for block i (6 x 6 array).	Pa
E	Young's modulus.	Pa
$\underline{f}$	Load vector in global solution equations ($6n$ x 1 array).	
$\underline{f}^i$	Load vector for block i (6 x 1 array).	
$\underline{f}'^i$	Transformed load vector for block i (6 x 1 array).	N
f_r^i	r-th term in load vector of block i.	
(f_x, f_y)	Body force per unit volume.	N/m^3
(F_x, F_y)	Point load in x- and y-direction.	N/m
$\underline{\underline{F}}^i{}_{int}$	Element internal force array (6 x 1 array).	N/m
$\widetilde{F}$	Friction force at block contact.	N/m
g	Acceleration due to gravity.	m/s^2
g	Displacement vector for contacts and bolts (6 x 1 array).	

Symbol	Description	Units
g0	Maximum incremental block displacement ratio.	
g1	Upper limit of incremental time interval.	s
g2	Contact penalty value.	N/m
k01	Dynamic or static analysis control parameter.	
$\underline{\underline{K}}$	Global stiffness matrix in solution equations ($6n$ x $6n$ array).	
$\underline{\underline{K}}^{ij}$	Stiffness matrix for interaction of blocks i and j (6 x 6 array).	
$\underline{\underline{K}}^{*ij}$	Transformed block stiffness matrix (6 x 6 array).	N/m
$K_{rs}{}^{ij}$	Single term in $\underline{\underline{K}}^{ij}$.	
l	Length of rock bolt, or block edge.	m
M	Mass per unit area.	kg/m^2
n	Number of blocks in a problem.	
p_b	Stiffness of rock bolt.	N/m
p_c	Contact penalty value.	N/m
p_d	Prescribed displacement penalty value.	N/m
$\underline{\underline{Q}}^{i}$	Transformation matrix for block i (6 x 6 array).	
r_0	Rotation of block centroid.	radians
$\underline{\underline{R}}^{i}$	Inverse of transformation matrix for block i (6 x 6 array).	
(R_s, R_n)	Shear and normal reaction forces at block contact.	N/m
S	Integer term for contact penetration calculation.	m^2
S^i	Area of block i.	m^2
$S_x{}^i$, $S_y{}^i$	First moments of area of block i.	m^3
$S_{xx}{}^i$, $S_{yy}{}^i$, $S_{xy}{}^i$	Second moments of area of block i.	m^4
t	Time.	s
Δt	Incremental time interval.	s
$\underline{\underline{T}}^i(x, y)$	Displacement matrix for point (x, y) of block i (2 x 6 array).	
$\underline{u}^i$	Displacement array for block i (6 x 1 array).	m
$u_r{}^i$	Single term in the displacement array of block i.	m
(u, v)	Displacement of a point.	m
(u_0, v_0)	Displacement of block centroid.	m
(u_m, v_m)	Prescribed displacement of a point.	m
v_i	Impact velocity in contact simulation.	m/s
$\underline{v}^i(t)$	Velocity array of block i at time t (6 x 1 array).	
(x, y)	Global co-ordinate system.	m
$(\bar{x}, \bar{y})$	Local block co-ordinate system.	m
(x_0, y_0)	Co-ordinates of block centroid.	
$\underline{\varepsilon}^i(x,y)$	Strain at point (x, y) of block i (3 x 1 array).	

Symbol	Description	Units
$(\varepsilon_x^0, \varepsilon_y^0, \gamma_{xy}^0)$	Strain at centroid of block.	
ε_x	Strain in x-direction.	
ε_y	Strain in y-direction.	
$\varepsilon_{x,x}$ $\varepsilon_{x,y}$ $\varepsilon_{y,x}$ $\varepsilon_{y,y}$	Rate of variation of strain.	m^{-1}
Φ	Angle of friction of discontinuity.	radians
γ_{xy}	Engineering shear strain.	
$\gamma_{xy,x}$ $\gamma_{xy,y}$	Rate of variation of shear strain in x- and y-direction.	m^{-1}
λ^*_k	Lagrange multiplier.	N
ν	Poisson's Ratio.	
Π	Total potential energy of system.	Nm
$\underline{\sigma}^i(x,y)$	Stress at point (x,y) of block i (3 x 1 array).	Pa
$(\sigma_x^0, \sigma_y^0, \tau_{xy}^0)$	Stress at centroid of block.	Pa
$\underline{\sigma_0}$	Initial stress in block (6 x 1 array).	Pa
σ_{VM}	Von Mises stress.	Pa
σ_x	Stress in x-direction.	Pa
σ_y	Stress in y-direction.	Pa
τ_{xy}	Shear stress.	Pa
(ξ, η)	Element co-ordinate system in FEM.	

Chapter 1

INTRODUCTION

The prediction of failure in rock represents a special range of problems in civil engineering. Most *in situ* rock is intersected by discontinuities such as faults, cracks, shear joints, tension joints, cleavage, and bedding planes. These discontinuities are often sufficiently prevalent that the rock mass actually resembles a system of separate and interlocking blocks, rather than a continuous medium. Figure 1.1 shows an example of a rock mass in the Table Mountain series, which contains many discontinuities.

Discontinuities may be thought of loosely as cracks or flaws within the rock. They have a range of geological origins and physical properties, but they can generally be defined as a plane across which compression forces can be transferred, but which has reduced or zero strength in shear and tension.

An important observation with regard to discontinuities is that they are generally not random, but are found in 'joint sets', with all of the discontinuities in a set having similar orientations and properties.

A continuous medium, or continuum, is a homogeneous body which will move and deform as a single unit when loading is applied. Failure is a result of the stresses in a part of the body increasing to a level greater than the strength of the material of which the body is composed. A discontinuous medium, or discontinuum, is a body containing discontinuities which may or may not divide the body into a number of separate blocks. Deformation or failure is generally a result of sliding or separation along

Figure 1.1: Devil's Peak, South Africa, with the University of Cape Town in the foreground.

these discontinuities, rather than failure of intact material in the rock blocks.

Early attempts [Blake, 1969; Yu *et al.*, 1968; Wang and Sun, 1970] to model the failure of rock masses by the Finite Element Method of analysis (FEM) were made under the assumption that the rock behaved as an isotropic elastic continuum. This type of analysis is of limited usefulness, because the presence of the discontinuities within the rock mass cannot be modelled except by using a general 'representative stiffness' to define the stiffness of the rock mass as a whole. Apart from the difficulties associated with estimating this representative stiffness, a weakness of the method is that the mode of deformation is independent of the orientation of the joints, whereas jointed rock masses are generally strongly anisotropic.

Elastic analyses determine stresses in a body, but do not model failure. An alternative approach is to treat the material as an orthotropic elastoplastic solid in the way that sand is typically modelled in FEM analyses. When this method is applied to jointed rock, it is based on the assumption that the blocks are relatively small, and that the discontinuities are sufficiently regular in orientation that the rock mass as a whole has a general anisotropic strength. Once again, this strength must be estimated in some manner. In hard rock, the strength of the intact rock is often one or two orders of magnitude higher than that of the rock mass [Hoek and Bray, 1981]. A further problem with this approach is that often the blocks are of a size comparable to the typical lengths of the excavation or structure, and in such cases, the movement of a single block will precipitate failure. This phenomenon cannot be simulated by manipulating the material model.

In order to effectively model jointed rock masses it is therefore necessary to be able to consider the effect of individual discontinuities on the strength and stiffness of the body as a whole.

In simple problems involving only a few discontinuities, modes of failure and factors of safety can be determined by means of hand calculations, and methods are well developed for solving these types of problems [e.g. Goodman and Shi, 1985]. However, with more complex geometries involving many discontinuities, it rapidly becomes impossible to find a solution by hand calculation, and a computer based numerical method is necessary.

A direct approach to modelling discontinuities within the Finite Element method is the use of joint and contact elements. Goodman *et al.* [1968] carried out pioneering research into the development of joint elements to allow splitting and separating along discontinuities within the Finite Element mesh. They were able to solve problems containing several hundred discontinuities by this method.

At present, many Finite Element, Boundary Element and Finite Difference programs have the capability to model discontinuities by the use of interface elements. However,

these programs, which are essentially designed for continuum modelling, often have drawbacks in modelling discontinua. New contacts are often not recognised, large displacements may not be allowed, and the number of discontinuities may be limited by the logic of the solution procedure [Cundall and Hart, 1993]. The introduction of discontinua is also computationally expensive, increasing solution times or placing limits on the complexity of solvable systems. Because of these limitations, the development of a numerical tool specifically designed to model discontinuous systems is desirable.

The Distinct Element Analysis method (DEA) was developed by Cundall [1971]. This method represented a completely new approach to the numerical modelling of discontinuous systems. Whereas previous methods had modelled the rock mass as a continuum, and treated discontinuities as a special case, Cundall's method modelled deformation purely in terms of differential movements of blocks. The blocks themselves were treated as rigid bodies, and internal stresses were not considered. An explicit, time-marching scheme was used to solve the equations of motion and balance of forces directly. Where blocks were in contact, they were prevented from inter-penetrating (overlapping) by the imposition of contact forces.

Since the introduction of the original DEA method, it has been combined with FEM. This mixed formulation allows blocks to be sub-divided into Finite Element meshes which may deform, while the interaction of the blocks is in accordance with the DEA theory. A fracture mechanism has also been incorporated so that blocks may split along element boundaries.

Other methods have been developed specifically for modelling discontinua, also based on the original DEA concept. These include Modal and Momentum Exchange methods [see Williams *et al.*, 1985] and collectively these different tools are called discrete element methods.

The most recent discrete element method is Discontinuous Deformation Analysis (DDA), introduced by Gen Hua Shi [Shi, 1988]. This method is based on similar premises to DEA. The blocks are deformable, according to a simple first-order displacement function, but the primary mode of deformation and failure is again the relative displacement of blocks along discontinuities. The most important difference between DDA and other discrete element methods is that the DDA method uses the principle of total potential energy minimisation to obtain a step-wise solution, whereas other methods solve for equilibrium by balance of forces and the Newtonian laws of motion. In this respect, DDA bears a close similarity to FEM. Like DEA, the DDA method uses an explicit time-stepping solution scheme, without iterative solving. DDA can be used to obtain solutions for both static and dynamic systems, and it solves for the stresses and strains within the rock blocks, as well as their displacements.

The Discontinuous Deformation Analysis method is the subject of this report, and a fuller description of the solution procedure is provided in the chapters that follow. The computer program used throughout this study is DDA Version 96 [1996] written by Shi. This program is described in Chapter 2. Chapter 3 sets out the time-stepping solution method behind Discontinuous Deformation Analysis in more detail.

In Chapter 4, the program is discussed and evaluated. A considerable volume of independent work has been carried out by various researchers since DDA was first introduced in 1988, and some of the extensions and modifications that have been made to the original method are described. The program is evaluated by the simulation of some elementary physical problems, and its performance in handling these problems is discussed.

An alternative method for sub-meshing of blocks is proposed in Chapters 5 to 7. This method allows a more accurate determination of the stresses and strains within certain, critical blocks; it also has the potential to model fracture and material non-linearity.

Chapter 2

OVERVIEW OF DISCONTINUOUS DEFORMATION ANALYSIS

An example of a DDA analysis is shown in Figure 2.1. A strong compression wave is introduced into a rock mass containing three tunnels, by cyclic displacement of the uppermost platten. The first view shows the rock mass early in the analysis, while the second shows the final result, after tunnel collapse. It will be seen that the method solves a problem not unlike a Finite Element type mesh in appearance. However, in DDA, each element is a polygonal block bounded by discontinuities, and sliding and separation can taken place along these discontinuities.

The DDA method adopts a time increment method of solution, generally arriving at the final solution by using a large number of small, progressive time-steps. Deformations of blocks are assumed to be first-order, and linear-elastic. It is an assumption of the method that the displacement of a block within a single time increment is small. With a large number of increments, however, the cumulative displacement of a block is effectively unlimited.

DDA can perform both static and dynamic analyses. In a static analysis, all forces on the blocks must be equilibrated at the end of each time step, while dynamic analyses take the velocity and momentum of the blocks into account to obtain a real-time incremental solution. In both cases, an explicit solution procedure is used without iterative solving. Iterative solving may be necessary, however, to control time-step intervals so as to ensure that incremental displacements remain small. Iterations may also be necessary if new contacts are formed within an increment, if blocks which were previously in contact separate, or to control inter-penetrations of blocks at contacts. In Section 3.4 the method of modelling contacts is described in more detail.

The program recognises new contacts between blocks automatically as the analysis progresses.

Only two-dimensional problems can be handled by existing software, but there are no fundamental difficulties involved in extending the theory to three dimensions.

A computer program has been developed by Gen-Hua Shi to carry out DDA analyses. The version of the program considered throughout this report is DDA Version 96 [Shi

1996], which incorporates some revisions to the original code, which was introduced in 1988.

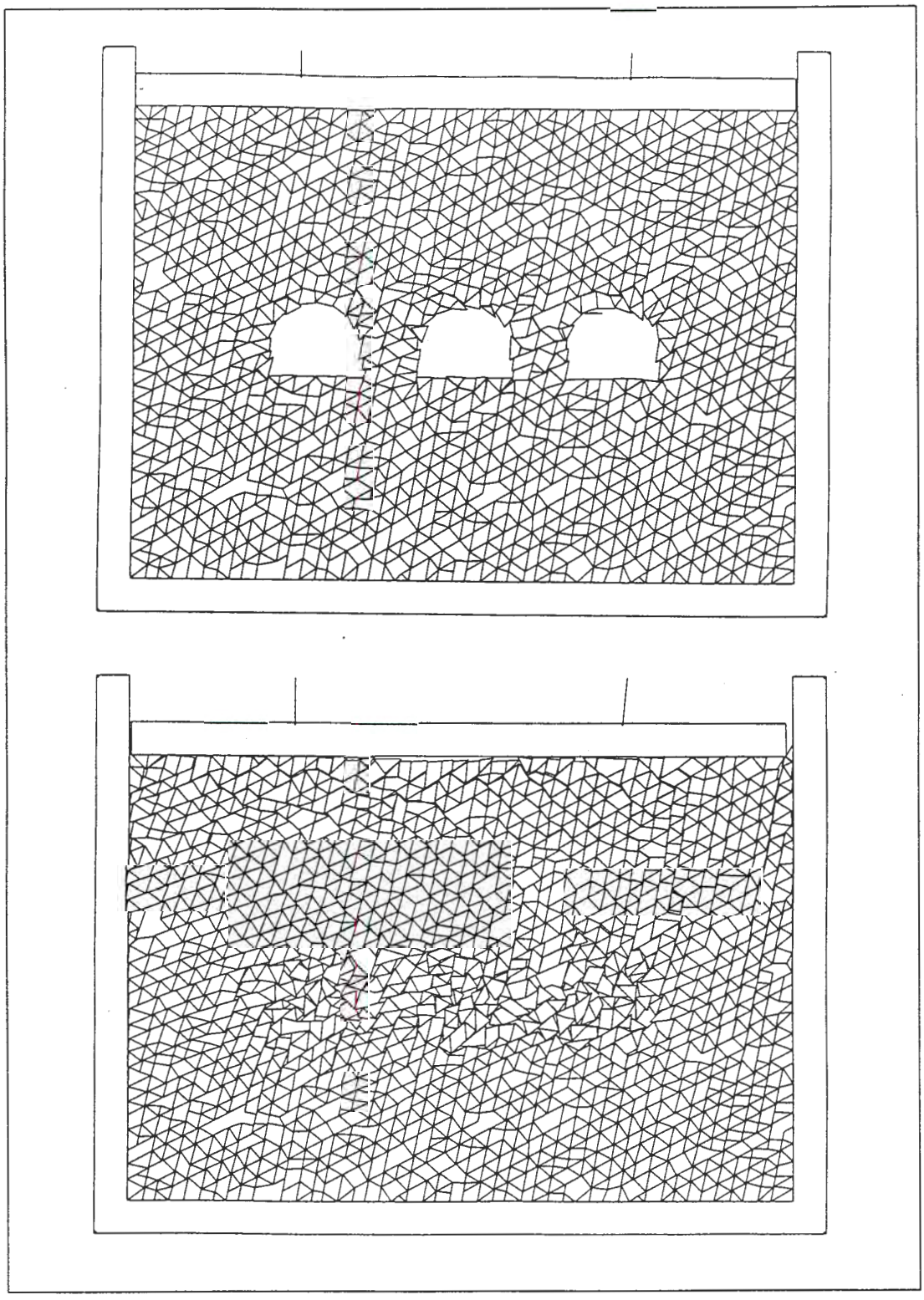


Figure 2.1: An example of a DDA analysis.

2.1 DDA Version 96 Computer Package

DDA Version 96 [1996] is a software package for carrying out 2-dimensional analyses of blocky systems using the DDA method. In this section, the DDA analysis procedure using DDA Version 96 is described. This chapter is not intended as a complete description either of the DDA method, or of the program. However, a step-wise description of the solution method provides a useful introduction to the DDA method as a whole.

The DDA Version 96 programs are written in the C programming language. Executable files are provided, the programs having been compiled using the NDP C|C++ compiler [1994] marketed by Microway, Inc.

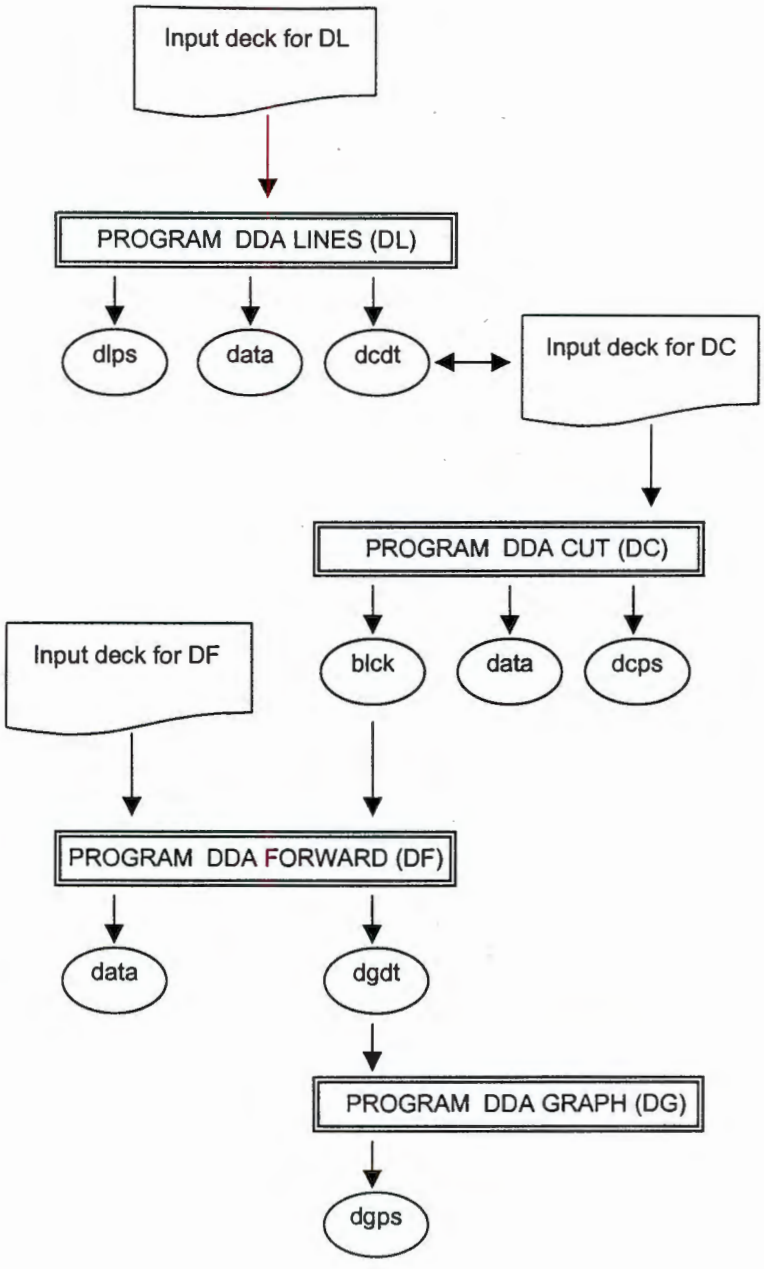
The package consists of four programs which are run consecutively, and these are described separately below. Programs DDA Lines and DDA Cut are pre-processors for setting up the geometry of the problem. Program DDA Forward performs the actual DDA analysis, and program DDA Graph is a post-processor for displaying the results of the analysis. Data is input by the user in the form of data files, or input decks, and the different programs pass information on to the next program in the sequence, also in the form of data files. Figure 2.2 shows the manner in which the programs, input decks and data files are inter-related.

2.1.1 Program DDA Lines

Program DDA Lines forms the first step in pre-processing a DDA problem. An input deck will have been prepared by the user, and this is read by DDA Lines. This input deck contains information on the problem boundary, the location and dimensions of any tunnels, and on any rock bolts, fixed lines, loading points and measured points in the solution domain. An example of an initial problem set-up is shown diagrammatically in Figure 2.3.

Fixed lines are a method of prescribing boundary conditions for a problem. Any block through which a fixed line passes is assigned two points along the line. The user may then specify that these points remain stationary during the analysis, thereby locking the block in position, or that they move in a specified manner.

A loading point is any position where point loads will be applied during the analysis. A measured point is any position in the problem domain for which the user wishes to obtain detailed information about displacements and stresses.



Data File Descriptions

dlps	*A print file generated by program DL showing initial problem set-up.*
dcdt	*The data file for program DC*
data	*A data file showing selected information on the program run*
blck	*The data file for program DF, giving block geometries and problem set-up*
dcps	*A print file generated by program DC showing pre-processed problem.*
dgdt	*The data file for program DG, giving the results of the analysis*
dgps	*A print file generated by program DG, showing the analysis results*

Figure 2.2: Inter-relationship of the DDA Version 96 program files [after Shi, 1988].

There is no practical way of determining the positions and dimensions of all discontinuities within an actual rock mass. However, discontinuities of the same geological origin are generally approximately parallel to each other. A geological investigation of a rock mass will yield parameters relating to these 'joint sets', such as average spacing, size and orientation of the discontinuities, and the degree of randomness.

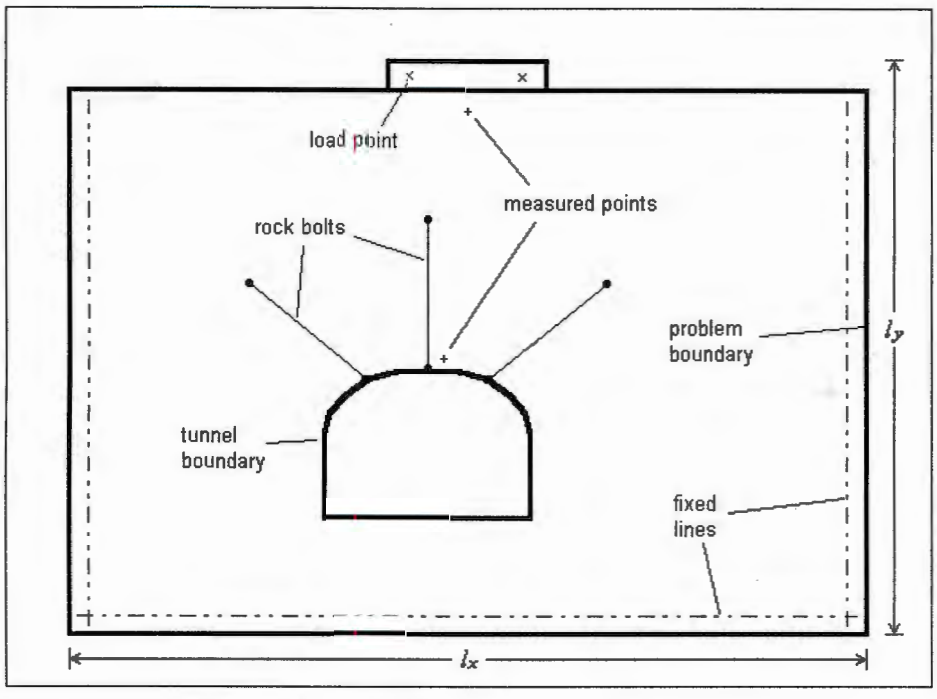


Figure 2.3: An initial set-up for a DDA analysis.

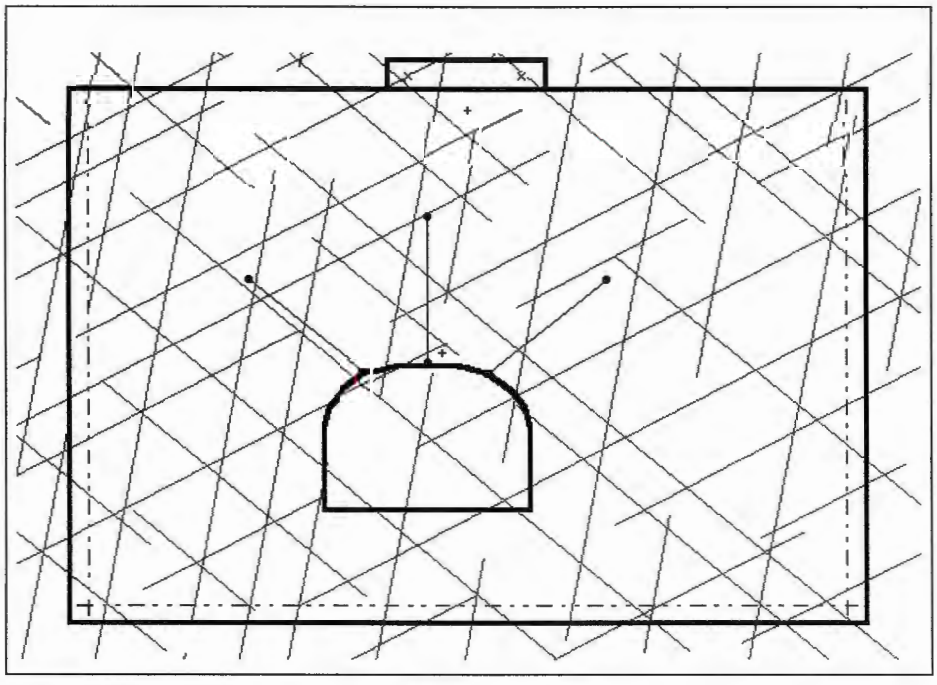


Figure 2.4: Randomised joint set generation by program DDA Lines.

These parameters are included in the input deck for program DDA Lines. The program then generates discontinuities through the rock mass, based on this information, as shown in Figure 2.4. Any individual known discontinuities can also be input separately. The data associated with all of these lines is saved as an output file, for use by program DDA Cut.

2.1.2 Program DDA Cut

The program DDA Cut forms the second phase of problem pre-processing. Starting from the output file from program DDA Lines, program DDA Cut generates the block system by forming all possible blocks defined by the discontinuity and boundary line segments. If joint set generation is not required, the user may choose to write an input deck for program DDA Cut directly, in which case program DDA Lines is not used.

The first stage in the block formation process is to truncate all joint lines which lie outside the boundary domain, or inside tunnels or other cavities. A block consists of a closed loop of line segments, and therefore any 'free end' of a discontinuity that ends within a block is irrelevant. Every discontinuity is therefore truncated at the first and last points where it crosses another discontinuity or boundary.

The program then follows around all loops and defines the blocks by their edge segments. This process is described in Shi [1988]. The result is as shown in Figure 2.5, where the problem is now defined in terms of the blocks, rather than the discontinuities and boundaries. Any block through which a fixed line passes is assigned two fixed points to control its displacement during the analysis. Finally, it is determined in which block the end points of rock anchors and loading and measured points lie.

All of the information for the pre-processed problem is stored in an output file, for use by the program DDA Forward.

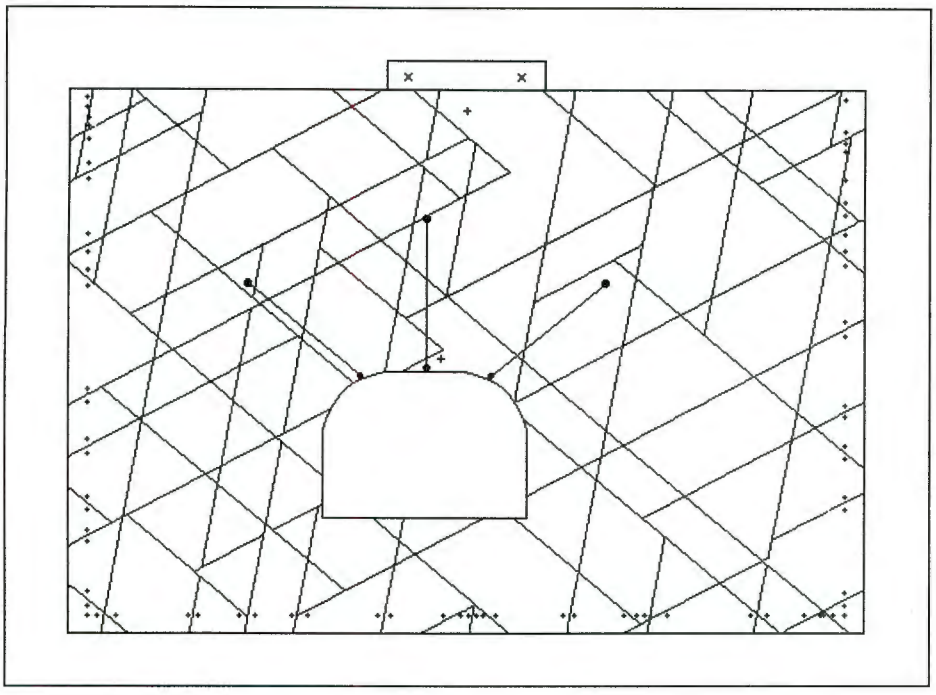


Figure 2.5: The problem domain when pre-processing is complete.

2.1.3 Program DDA Forward

Program DDA Forward carries out the actual DDA analysis of the problem. It requires the data file generated by program DDA Cut, and it also requires a further input deck, created by the user. This input deck contains information on the material moduli and loading and boundary conditions. It also specifies whether the analysis is to be static or dynamic. The DDA program will by default control the time step intervals and contact spring penalty value in order to optimise the analysis run. However, the user may specify a fixed penalty value, and an upper limit for the time step interval. These are also contained in the input deck.

A flow-chart showing the analysis procedure is shown in Figure 2.6. The steps are labelled sequentially, and are briefly described below.

1. The information from the data file produced by program DC and from the user input deck is read, and the problem is set up.
2. An initial contact penalty value is determined. The DDA program controls the stiffness of contacts as the analysis proceeds, unless the user specifies a fixed value. If the stiffness is too great, then accuracy is lost in the modelling of contacts. If the stiffness is too small, then inter-block penetrations are too large.
3. The first time increment begins. The DDA program automatically controls the time interval used in the time-stepping procedure to optimise the analysis run. If the time interval is too small, then a large number of increments are required to

complete an analysis. On the other hand, if the time interval is too large, then incremental displacements become large also, and are not adequately modelled by the program, which uses first order displacement functions.

4. The program searches for blocks that are likely to be in contact and sets up penalty functions to control these contacts.
5. The solution equations are set up for the increment.
6. The contact penalty terms are added to the solution equations.
7. The system is solved, and the displaced positions of the blocks at the end of the time step are obtained.
8. The initial contact assumptions are reviewed. In particular it is necessary to check if there are any blocks which are inter-penetrating without contact control. In addition, the penetrations at contacts are checked, and if they are too great or too small the penalty function value is adjusted.
9. The maximum incremental displacement of the blocks is determined.
10. If the contact assumptions were found to be wrong, the system is re-solved with revised contacts.
11. In the event that it requires more than six iterations to find the correct contact assumptions, the time interval is reduced, and the iteration calculations repeated.
12. If the incremental block displacements are too great, then the accuracy of the analysis is reduced. Therefore, if the maximum displacement ratio is too large, the time interval is reduced, and the increment re-started.
13. Inter-block penetrations were determined at step 8. If they are too large, then the increment is repeated with the revised penalty value, which was also calculated in step 8.
14. The block vertex positions are updated.
15. The stresses and velocities of the blocks at the end of the increment are determined for use in the next time increment.
16. Simplex integration is carried out for each block, in preparation for the next time increment. Simplex integration is a method of determining the moments of area of the blocks, from the vertex co-ordinates. These simplexes are used in determining the solution equations (see Section 3.3.6).
17. The increment is now complete, and the procedure is repeated from step 3 for the next increment, until the analysis is complete.

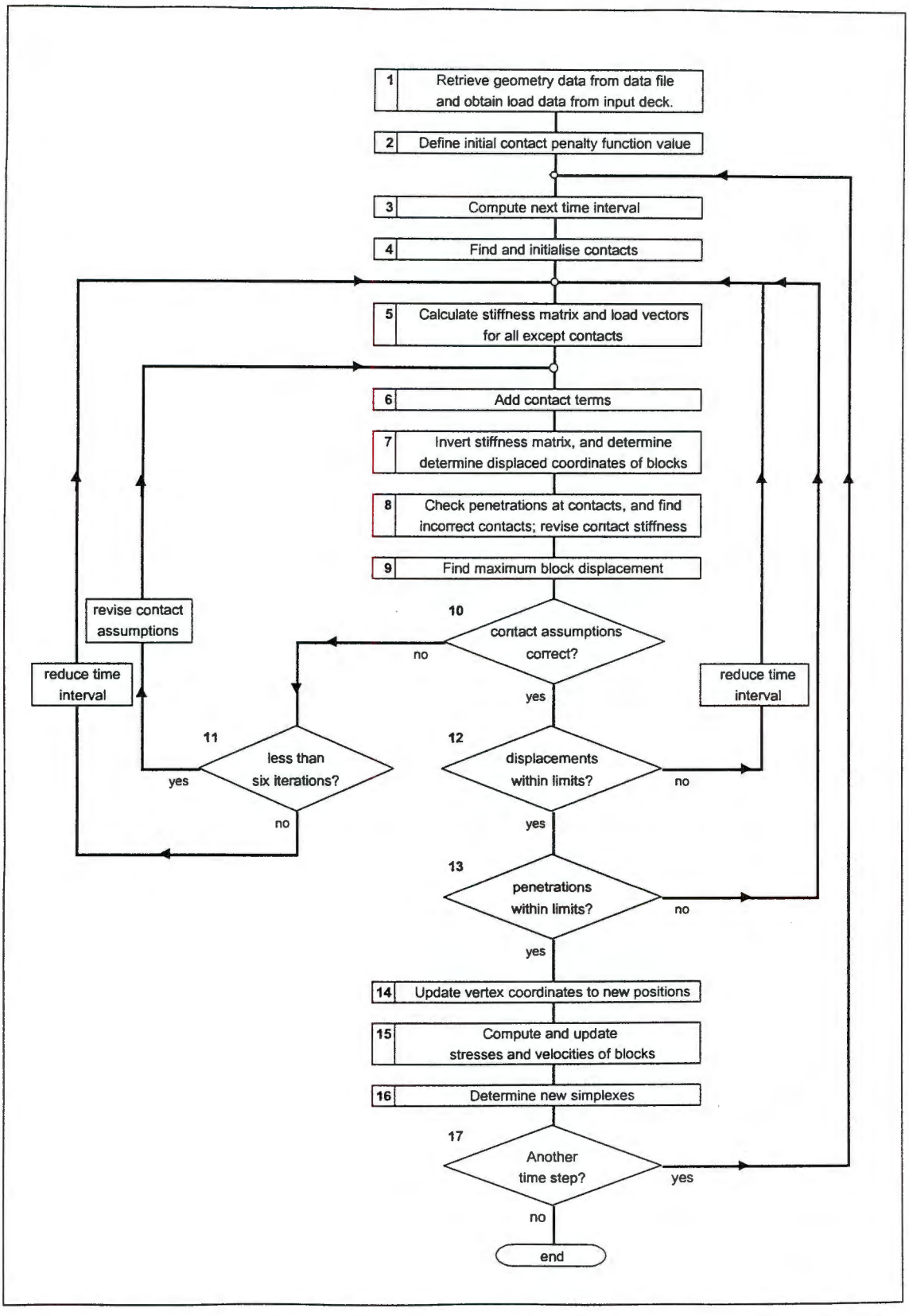


Figure 2.6: A flow-chart for program DDA Lines

The program produces two output files. A data file contains selected information on the analysis process, and a further data file is produced as an input file for program DDA Graph.

2.1.4 Program DDA Graph

Program DDA Graph displays the results of the DDA analysis in the form of a 'slide show' of the displaced positions of the blocks. Only selected time steps are shown, and the information is retrieved from a data file which was produced by program DDA Forward for this purpose. A print file is also produced.

Chapter 3

THE DDA FORWARD ANALYSIS METHOD

The DDA Forward program creates and solves a system of simultaneous equations in order to achieve equilibrium at the end of each time-step. In this chapter, these equations are described. The description is based on the work of Gen-Hua Shi [1988].

3.1 Co-ordinate System and Solution Variables

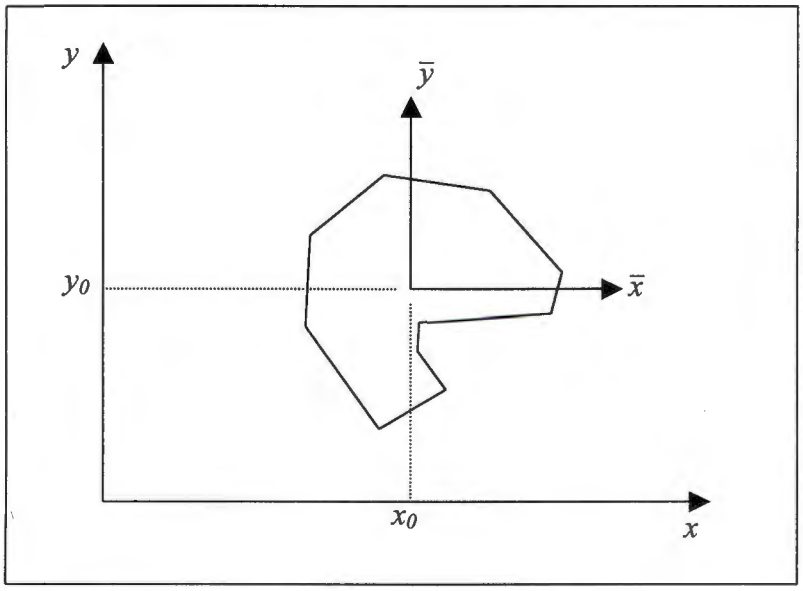


Figure 3.1: A single polygonal block.

A single polygonal block, block i, is depicted in Figure 3.1. A local block co-ordinate system is used, with the axes orientated parallel to the global system, but with the origin at the centroid of the block. In DDA, the deformation variables for each block are the rigid body displacements of the centroid of the block, u_0, v_0, r_0 (the horizontal and vertical displacements, and rotation, respectively), and the strains, ε_x, ε_y, γ_{xy}, which are assumed to be constant throughout the block. Using these parameters, the displacement (u, v) of any point (x, y) in the block can be determined from the equation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{\bar{y}}{2} \\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{\bar{x}}{2} \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \\ r_0 \\ \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \underline{\underline{T}}^i(x,y)\underline{d}^i \qquad (3.1)$$

where

$$\bar{x} = x - x_0$$
$$\bar{y} = y - y_0$$

and (x_0, y_0) are the co-ordinates of the centroid of the block.

Equation 3.1 can also be written in the form

$$u = u_0 + \varepsilon_x \bar{x} + (\gamma_{xy}/2 - r_0)\bar{y}$$
$$v = v_0 + (\gamma_{xy}/2 + r_0)\bar{x} + \varepsilon_y \bar{y}$$

which demonstrates that this is a complete first order displacement function for the block.

The advantage of using (u_0 v_0 r_0 ε_x ε_y γ_{xy}) as the deformation variables is that each of these terms has obvious physical meaning. Additionally, the energy equations obtained using these parameters are of the simplest form.

The stresses in the block for plane stress conditions can be determined from:

$$\begin{pmatrix} \sigma_x \\ \sigma_y \\ \tau_{xy} \end{pmatrix} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{pmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix} . \qquad (3.2)$$

3.2 Global Solution Equations

Static or dynamic equilibrium is maintained based on the principle of stationary potential energy. This principle states that a system that has been deformed under loading satisfies the equations of equilibrium if the total potential energy of the system remains unchanged for a small perturbation of any of its degrees of freedom. The total potential energy (TPE) of the system, Π, is defined as the sum of the internal strain energy of the system and the external work done by the loading.

In the DDA method, the degrees of freedom (d.o.f.) of each block in the system are the six block deformation variables described in Section 3.1. The principle of stationary potential energy may be written as

$$\frac{\partial \Pi}{\partial d_r} = 0 \qquad (3.3)$$

where d_r is any of the degrees of freedom of any block in the system.

An expression may be derived for the TPE of the system as a whole. This expression is of the form

$$\Pi = \frac{1}{2} d_r K_{rs} d_s - d_t f_t \quad . \qquad (3.4)$$

Substituting Equation 3.4 into Equation 3.3,

$$\frac{\partial \Pi}{\partial d_r} = K_{rs} d_s - f_r = \underline{0} \quad .$$

A system of simultaneous equilibrium equations is therefore built up with the deformation variables as unknowns. For a system of n blocks, the equilibrium equations are written in matrix form:

$$\begin{bmatrix} \underline{\underline{K}}^{11} & \underline{\underline{K}}^{12} & \underline{\underline{K}}^{13} & \cdots & \underline{\underline{K}}^{1n} \\ \underline{\underline{K}}^{21} & \underline{\underline{K}}^{22} & \underline{\underline{K}}^{23} & \cdots & \underline{\underline{K}}^{2n} \\ \underline{\underline{K}}^{31} & \underline{\underline{K}}^{32} & \underline{\underline{K}}^{33} & \cdots & \underline{\underline{K}}^{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ \underline{\underline{K}}^{n1} & \underline{\underline{K}}^{n2} & \underline{\underline{K}}^{n3} & \cdots & \underline{\underline{K}}^{nn} \end{bmatrix} \begin{pmatrix} \underline{d}^1 \\ \underline{d}^2 \\ \underline{d}^3 \\ \vdots \\ \underline{d}^n \end{pmatrix} = \begin{pmatrix} \underline{f}^1 \\ \underline{f}^2 \\ \underline{f}^3 \\ \vdots \\ \underline{f}^n \end{pmatrix}$$

or $\underline{\underline{K}}\,\underline{d} = \underline{f} \quad .$ (3.5)

In the above set of equations, each element $\underline{\underline{K}}^{ij}$ is a 6 x 6 sub-matrix, each element $\underline{d}^i$ is the deformation variables for block i, $(u_0 , v_0 , r_0 , \varepsilon_x , \varepsilon_y , \gamma_{xy})^T$, and each term $\underline{f}^i$ represents the loading on block i distributed to the six deformation variables.

Thus, $\underline{\underline{K}}$ is a $6n$ x $6n$ square matrix and is referred to as the stiffness matrix of the system. It is symmetric, and invertible. $\underline{d}$ and $\underline{f}$ are column vectors, each with $6n$ terms, and are named the deformation vector and the load vector, respectively.

The equilibrium equations are solved simultaneously by inverting the stiffness matrix, and then solving for the system deformation variables

$$\underline{d} = \underline{\underline{K}}^{-1} \underline{f} \quad . \tag{3.6}$$

3.3 Equilibrium of a Single Block

In this section the total potential energy of a single block is considered without interaction with other blocks. Because there is no interaction, the potential energy of the block is independent of the deformation variables of other blocks in the system. Therefore, for a block i, the only relevant terms in the global solution equations (Equation 3.5) are the block load vector, $\underline{f}^i$, and the diagonal element in the stiffness matrix, $\underline{\underline{K}}^{ii}$, where $\underline{\underline{K}}^{ii}$ is again a 6 x 6 matrix.

The potential energy of the block is composed of a sum of several separate components. These are the elastic strain of the block, Π_e, the initial stress at the start of the time increment, Π_σ, the external energy of point and volume loads, Π_p and Π_v, any prescribed displacements of the block, Π_m, and the forces of inertia for dynamic analyses, Π_i.

These components are described separately below. In each case, the potential energy is derived as a function of the deformation variables in the form of Equation 3.4, and then the stiffness matrix and load array terms are derived separately by differentiation:

$$\Pi^i = \frac{1}{2} d_r^i K_{rs}^{ii} d_s^i - d_t^i f_t^i$$

therefore,

$$\frac{\partial^2}{\partial d_r^i . \partial d_s^i}\left(\frac{1}{2} d_r^i K_{rs}^{ii} d_s^i\right) \Rightarrow \underline{\underline{K}}^{ii}$$

$$-\frac{\partial}{\partial d_r^i}\left(-d_r^i f_r\right) \qquad \Rightarrow \underline{f}^i \quad .$$

3.3.1 Elastic strain

The strain energy Π_e of a single block under elastic strain is defined as

$$\Pi_e = \int\int \frac{1}{2}(\varepsilon_x \sigma_x + \varepsilon_y \sigma_y + \gamma_{xy} \tau_{xy}).dx.dy \tag{3.7}$$

where integration is over the area of the block. For plane stress conditions, using Equation 3.2, this becomes

$$\Pi_e = \int\int \frac{1}{2}\begin{pmatrix}\varepsilon_x & \varepsilon_y & \gamma_{xy}\end{pmatrix}\begin{pmatrix}\sigma_x \\ \sigma_y \\ \tau_{xy}\end{pmatrix}.dx.dy$$

$$\Pi_e = \frac{1}{2}\left(\frac{E}{1-\nu^2}\right)\int\int \begin{pmatrix}\varepsilon_x & \varepsilon_y & \gamma_{xy}\end{pmatrix}\begin{bmatrix}1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2}\end{bmatrix}\begin{pmatrix}\varepsilon_x \\ \varepsilon_y \\ \gamma_{xy}\end{pmatrix}.dx.dy$$

$$\Pi_e = \frac{1}{2}\int\int \underline{d}^{i^T}\, \underline{\underline{E}}\, \underline{d}^i .dx.dy$$

where $\underline{\underline{E}}$ is defined as

$$\underline{\underline{E}} = \frac{E}{1-\nu^2}\begin{bmatrix}0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \nu & 0 \\ 0 & 0 & 0 & \nu & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2}\end{bmatrix} . \tag{3.8}$$

Stresses and strains are constant in the block, so that

$$\Pi_e = \frac{1}{2} S^i \underline{d}^{i^T}\, \underline{\underline{E}}^i\, \underline{d}^i \tag{3.9}$$

where S^i is the area of the block.

Derivatives are computed to minimise the strain energy Π_e

$$K_{rs}^{ii} = \frac{1}{2} S^i \frac{\partial^2}{\partial d_r^i . \partial d_s^i} \left(\underline{d}^{iT} \underline{\underline{E}} \ \underline{d}^i \right) = S^i \underline{\underline{E}}$$

$$S^i \underline{\underline{E}} \Rightarrow \underline{\underline{K}}^{ii} \qquad (3.10)$$

The 6 x 6 matrix of Equation 3.10 is added to the submatrix $\underline{\underline{K}}^{ii}$ in the global solution equations (Equation 3.5).

3.3.2 Initial stress

If the block is under an initial stress of (σ_x^o, σ_y^o, τ_{xy}^o), the potential energy of the initial stress is

$$\Pi_\sigma = \int\int \left(\varepsilon_x \sigma_x^{\,o} + \varepsilon_y \sigma_y^{\,o} + \gamma_{xy} \tau_{xy}^{\,o} \right) dx.dy \qquad (3.11)$$

$$\Pi_\sigma = S^i \underline{d}^{iT} \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_x^0 \\ \sigma_y^0 \\ \tau_{xy}^0 \end{pmatrix}$$

or $$\Pi_\sigma = S^i \underline{d}^{iT} \underline{\sigma}_0^i \qquad (3.12)$$

where $$\underline{\sigma}_0^i = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sigma_x^0 \\ \sigma_y^0 \\ \tau_{xy}^0 \end{pmatrix}$$

The derivative of Equation 3.12 is derived to obtain the term $\underline{f}^i$:

$$f_r^i = -\frac{\partial \Pi_\sigma}{\partial d_r^i} = -S^i \frac{\partial}{\partial d_r^i} \left(\underline{d}^{iT} \underline{\sigma}_0^i \right) = S^i \underline{\sigma}_{0r}^i$$

$$-S^i \underline{\sigma}_o^i \Rightarrow \underline{f}^i \qquad (3.13)$$

This 6 x 1 array is added to $\underline{f}^i$ in the load vector of the global solution equations (Equation 3.5).

3.3.3 Point loading

If a point load (F_x, F_y) acts at a point (x, y) in block i, the potential energy of the load is

$$\Pi_p = -(F_x\ \mathrm{u} + F_y\ \mathrm{v}) \tag{3.14}$$

$$\Pi_p = -(\mathrm{u}\quad \mathrm{v})\begin{pmatrix} F_x \\ F_y \end{pmatrix} \quad .$$

Substituting Equation 3.1 into this,

$$\Pi_p = -\underline{d}^{iT}\,\underline{\underline{T}}^i(x,y)^T\begin{pmatrix} F_x \\ F_y \end{pmatrix} \quad . \tag{3.15}$$

To minimise Π_p, the derivatives are computed:

$$f_r^i = -\frac{\partial \Pi_p}{\partial d_r^i} = \frac{\partial}{\partial d_r^i}\left(\underline{d}^{iT}\,\underline{\underline{T}}^i(x,y)^T\begin{pmatrix} F_x \\ F_y \end{pmatrix}\right) = \underline{\underline{T}}^i(x,y)^T\begin{pmatrix} F_x \\ F_y \end{pmatrix} \quad .$$

f_r^i , r=1,.., 6, forms a 6 x 1 submatrix:

$$\underline{\underline{T}}^i(x,y)^T\begin{pmatrix} F_x \\ F_y \end{pmatrix} \Rightarrow \underline{f}^i \tag{3.16}$$

which is added to the submatrix $\underline{f}^i$ in the global solution equations (Equation 3.5).

3.3.4 Volume loading

If the entire block is acted on by a constant body force (f_x , f_y), then the potential energy is:

$$\Pi_v = -\int\int (f_x \mathrm{u} + f_y \mathrm{v}) dx.dy \tag{3.17}$$

where (u , v) is the displacement of a point (x , y) in the block.

$$\Pi_v = -\underline{d}^{iT} \int\int \underline{\underline{T}}^i (x, y)^T .dx.dy \begin{pmatrix} f_x \\ f_y \end{pmatrix} \quad .$$

The term $\int\int \underline{\underline{T}}^i (x, y).dx.dy$ is evaluated separately:

$$\int\int \underline{\underline{T}}^i (x, y)^T = \begin{bmatrix} S^i & 0 \\ 0 & S^i \\ -S_y^i & S_x^i \\ S_x^i & 0 \\ 0 & S_y^i \\ \frac{S_y^i}{2} & \frac{S_x^i}{2} \end{bmatrix} \tag{3.18}$$

where

$$S^i = \int\int .dx.dy$$

$$S_x^i = \int\int \bar{x}.dx.dy$$

$$S_y^i = \int\int \bar{y}.dx.dy \quad .$$

The term S^i is simply the area of the block, and because $(\bar{x}, \bar{y})$ is relative to the centroid of the block, $S_x^i = S_y^i = 0$ so that Equation 3.17 becomes

$$\Pi_v = -\underline{d}^{iT} \begin{pmatrix} f_x S^i \\ f_y S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad . \tag{3.19}$$

Minimising the potential energy by taking the derivatives,

$$f_r^i = -\frac{\partial \Pi_v(0)}{\partial d_r^i} = \frac{\partial}{\partial d_r^i}(\underline{d}^{iT}\begin{pmatrix} f_x S^i \\ f_y S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix})$$

which forms a 6 x 1 submatrix

$$\begin{pmatrix} f_x S^i \\ f_y S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \underline{f}^i \qquad (3.18)$$

and this is added to the term $\underline{f}^i$ in the global solution equation (Equation 3.5).

3.3.5 Prescribed displacements at a point

It may be necessary to fix a point in a block as a boundary condition of the problem, or to stipulate that the point moves by a prescribed amount.

If a point (x, y) is assigned a displacement of (u_m, v_m), this can be incorporated into the DDA method by placing a stiff spring of stiffness p_d between the point (x, y) and the stipulated displacement position.

The strain energy of the spring, Π_m, is

$$\Pi_m = \frac{p_d}{2}\left((u - u_m)^2 + (v - v_m)^2\right) \quad . \qquad (3.21)$$

$$\Pi_m = \frac{p_d}{2}\left((u - u_m \quad v - v_m)\begin{pmatrix} u - u_m \\ v - v_m \end{pmatrix}\right)$$

$$\Pi_m = \frac{p_d}{2}(u \quad v)\begin{pmatrix} u \\ v \end{pmatrix} - p_d(u \quad v)\begin{pmatrix} u_m \\ v_m \end{pmatrix} + \frac{p_d}{2}(u_m \quad v_m)\begin{pmatrix} u_m \\ v_m \end{pmatrix} \quad .$$

Because $\begin{pmatrix} u \\ v \end{pmatrix} = \underline{\underline{T}}^i(x,y)\underline{d}^i$ and $\begin{pmatrix} u & v \end{pmatrix} = \underline{\underline{T}}^i(x,y)^T \underline{d}^{iT}$,

$$\Pi_m = \frac{p_d}{2}\underline{d}^{iT}\underline{\underline{T}}^i(x,y)^T\underline{\underline{T}}^i(x,y)\underline{d}^i - p_d\underline{d}^{iT}\underline{\underline{T}}^i(x,y)^T\begin{pmatrix} u_m \\ v_m \end{pmatrix} + \frac{p_d}{2}\begin{pmatrix} u_m & v_m \end{pmatrix}\begin{pmatrix} u_m \\ v_m \end{pmatrix} \quad . \quad (3.22)$$

Minimising Π_m by taking derivatives,

$$K_{rs}^{ii} = \frac{\partial^2}{\partial d_r^i \partial d_s^i}\frac{p_d}{2}\left(\underline{d}^{iT}\underline{\underline{T}}^i(x,y)^T\underline{\underline{T}}^i(x,y)\underline{d}^i\right)$$

$$f_r^i = -\frac{\partial}{\partial d_r^i}\left(-p_d\underline{d}^{iT}\underline{\underline{T}}^i(x,y)^T\begin{pmatrix} u_m \\ v_m \end{pmatrix}\right)$$

where K_{rs}^{ii} forms a 6 x 6 submatrix which is added to the stiffness matrix of Equation 3.5:

$$p_d\underline{\underline{T}}^i(x,y)^T\underline{\underline{T}}^i(x,y) \Rightarrow \underline{\underline{K}}^{ii} \quad (3.23)$$

and f_r^i forms a 6 x 1 submatrix which is added to the force vector of Equation 3.5:

$$p_d\underline{\underline{T}}^i(x,y)^T\begin{pmatrix} u_m \\ v_m \end{pmatrix} \Rightarrow \underline{f}^i \quad . \quad (3.24)$$

3.3.6 Forces of inertia

In dynamic analyses, the velocities at the beginning of the time-step are taken into account in determining equilibrium. In static analyses, the velocities are reset to zero at the beginning of each time increment. This is the only difference between static and dynamic analyses in the DDA method.

Defining $u(x,y,t)$ and $v(x,y,t)$ as the time dependent displacement of any point in the block, the force of inertia per unit area is

$$\begin{pmatrix} f_x(x,y,t) \\ f_y(x,y,t) \end{pmatrix} = -M\frac{\partial^2}{\partial t^2}\begin{pmatrix} u(x,y,t) \\ v(x,y,t) \end{pmatrix} = -M\frac{\partial^2}{\partial t^2}\left(\underline{\underline{T}}^i(x,y)\ \underline{d}^i(t)\right) \quad (3.25)$$

where M is the mass per unit area of the block.

The potential energy of the inertia force of block i is defined by

$$\Pi_i = -\int\int \begin{pmatrix} u(x,y,t) & v(x,y,t) \end{pmatrix} \begin{pmatrix} f_x(x,y,t) \\ f_y(x,y,t) \end{pmatrix} .dx.dy \qquad (3.26)$$

$$\Pi_i = M \int\int \begin{pmatrix} u(x,y,t) & v(x,y,t) \end{pmatrix} \underline{\underline{T}}^i(x,y) \left(\frac{\partial^2}{\partial t^2} \underline{d}^i(t) \right) .dx.dy \quad .$$

Define the displacements at the beginning of the time-step to be zero, i.e. $\underline{d}^i(0)=\underline{0}$; the time interval of the step to be Δt; and the displacement at the end of the step to be $\underline{d}^i(\Delta t) = \underline{d}^i$. Assuming that acceleration is constant within the time-step,

$$\underline{d}^i = \underline{d}^i(\Delta t) = \underline{d}^i(0) + \Delta t \frac{\partial}{\partial t} \left(\underline{d}^i(0) \right) + \frac{\Delta t^2}{2} \frac{\partial^2}{\partial t^2} \left(\underline{d}^i(0) \right)$$

$$\frac{\partial^2}{\partial t^2} \underline{d}^i(0) = \frac{\partial^2}{\partial t^2} \underline{d}^i(\Delta t) = \frac{2}{\Delta t^2} \underline{d}^i - \frac{2}{\Delta t} \frac{\partial}{\partial t} \underline{d}^i(0) = \frac{2}{\Delta t^2} \underline{d}^i - \frac{2}{\Delta t} \underline{v}^i(0) \qquad (3.27)$$

where $\underline{v}^i(0) = \frac{\partial}{\partial t} \left(\underline{d}^i(0) \right)$.

Therefore, at time Δt, the potential energy is

$$\Pi_i = \underline{d}^{iT} \int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y).dx.dy \left(\frac{2M}{\Delta t^2} \underline{d}^i - \frac{2M}{\Delta t} \underline{v}^i(0) \right) \quad . \qquad (3.28)$$

The term $\int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y).dx.dy$ must be evaluated to obtain an analytical solution:

$$\underline{\underline{T}}^i(x,y)^T\underline{\underline{T}}^i(x,y)=\begin{bmatrix}1 & 0\\ 0 & 1\\ -\bar{y} & \bar{x}\\ \bar{x} & 0\\ 0 & \bar{y}\\ \frac{\bar{y}}{2} & \frac{\bar{x}}{2}\end{bmatrix}\begin{bmatrix}1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{\bar{y}}{2}\\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{\bar{x}}{2}\end{bmatrix}$$

$$=\begin{bmatrix}1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{\bar{y}}{2}\\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{\bar{x}}{2}\\ -\bar{y} & \bar{x} & \bar{y}+\bar{x} & -\overline{yx} & \overline{xy} & -\frac{\bar{y}^2}{2}+\frac{\bar{x}^2}{2}\\ \bar{x} & 0 & -\overline{yx} & \bar{x}^2 & 0 & \frac{\overline{xy}}{2}\\ 0 & \bar{y} & \overline{xy} & 0 & \bar{y}^2 & \frac{\overline{yx}}{2}\\ \frac{\bar{y}}{2} & \frac{\bar{x}}{2} & -\frac{\bar{y}^2}{2}+\frac{\bar{x}^2}{2} & \frac{\overline{xy}}{2} & \frac{\overline{yx}}{2} & \frac{\bar{y}^2}{4}+\frac{\bar{x}^2}{4}\end{bmatrix} \quad . \quad (3.29)$$

Define:

$$S^i=\int\int .dx.dy$$
$$S^i_x=\int\int \bar{x}.dx.dy$$
$$S^i_y=\int\int \bar{y}.dx.dy$$
$$S^i_{xx}=\int\int \bar{x}^2.dx.dy$$
$$S^i_{yy}=\int\int \bar{y}^2.dx.dy$$
$$S^i_{xy}=\int\int \overline{xy}.dx.dy \quad . \quad (3.30)$$

These terms are calculated direcly for each block from the vertex co-ordinates by simplex integration, which is described in Shi [1988]. The integrals are over the area of the block. Because the local co-ordinate system has its origin at the centroid of the block, $S^i_x=S^i_y=0$

Using Equations 3.30, the integral of Equation 3.29 becomes:

$$\int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y).dx.dy = \begin{bmatrix} S^i & 0 & 0 & 0 & 0 & 0 \\ 0 & S^i & 0 & 0 & 0 & 0 \\ 0 & 0 & S^i_{xx}+S^i_{yy} & -S^i_{xy} & S^i_{sy} & \frac{S^i_{xx}-S^i_{yy}}{2} \\ 0 & 0 & -S^i_{xy} & S^i_{xx} & 0 & \frac{S^i_{xy}}{2} \\ 0 & 0 & S^i_{xy} & 0 & S^i_{yy} & \frac{S^i_{xy}}{2} \\ 0 & 0 & \frac{S^i_{xx}-S^i_{yy}}{2} & \frac{S_{xy}}{2} & \frac{S_{xy}}{2} & \frac{S^i_{xx}+S^i_{yy}}{4} \end{bmatrix} \quad . \quad (3.31)$$

The potential energy function of Equation 3.28 is minimised with respect to $\underline{d}^i$:

$$K^{ii}_{rs} = \frac{\partial^2}{\partial d^i_r \partial d^i_s}\left(\underline{d}^{iT}\left(\int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y).dx.dy\right)\frac{2M}{\Delta t^2}\underline{d}^i\right)$$

$$f^i_r = -\frac{\partial}{\partial d^i_r}\left(-\underline{d}^{iT}\left(\int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y).dx.dy\right)\frac{2M}{\Delta t}\underline{\mathbf{v}}^i(0)\right) \quad .$$

These terms provide a 6 x 6 matrix and a 6 x 1 vector which are added to $\underline{\underline{K}}^{ii}$ and $\underline{f}^i$ of Equation 3.5 respectively:

$$\frac{2M}{\Delta t^2}\left(\int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y).dx.dy\right) \Rightarrow \underline{\underline{K}}^{ii} \quad (3.32)$$

$$\frac{2M}{\Delta t}\left(\int\int \underline{\underline{T}}^i(x,y)^T \underline{\underline{T}}^i(x,y)\right)\underline{\mathbf{v}}^i(0) \Rightarrow \underline{f}^i \quad . \quad (3.33)$$

The velocity at the end of the time step is calculated for use as $\underline{\mathbf{v}}^i(0)$ in the next iteration. Using Equation 3.27,

$$\underline{\mathbf{v}}^i(\Delta t) = \underline{\mathbf{v}}^i(0) + \Delta t.\frac{\partial^2}{\partial t^2}\underline{d}^i = \frac{2}{\Delta t}\underline{d}^i - \underline{\mathbf{v}}^i(0) \quad . \quad (3.34)$$

3.4 Block System Equilibrium

The derivations set out in Section 3.3 represent the minimisation of the internal energy of individual blocks under loading. In addition to this, the blocks within the system interact with each other. There are two situations where block interaction occurs in DDA. The most common is if two blocks are in contact with each other. The other situation is where two blocks are tied together by means of a rock bolt or anchor. The maintenance of equilibrium under these circumstances is the subject of this section.

3.4.1 Rock bolt connection

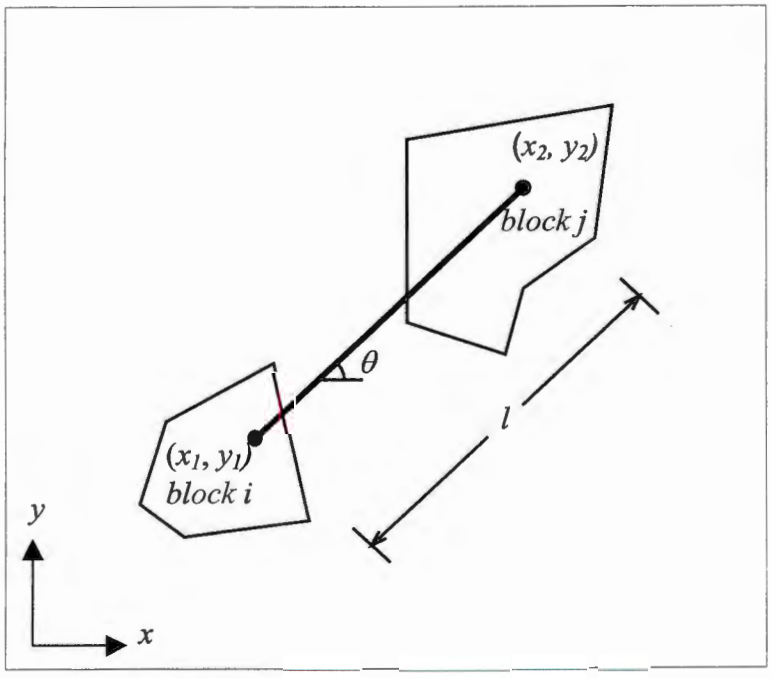


Figure 3.2: A rock bolt connecting blocks i and j.

Figure 3.2 shows a rock bolt connecting point (x_1, y_1) of block i to point (x_2, y_2) of block j. The bolt is of length l, lies at an angle θ to the x-axis, and has a stiffness of p_b.

The length of the bolt at the beginning of the time-step is given by

$$l = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad .$$

At the end of the time step, points (x_1, y_1) and (x_2, y_2) will have displaced by (u_1, v_1) and (u_2, v_2), where

$$(\mathrm{u}_1 \quad \mathrm{v}_1) = \underline{d}^{i^T} \underline{\underline{T}}^i (x_1, y_1)^T$$
$$(\mathrm{u}_2 \quad \mathrm{v}_2) = \underline{d}^{j^T} \underline{\underline{T}}^j (x_2, y_2)^T$$

and the change in length of the bolt, *dl*, is

$$dl = [(\mathrm{u}_1 - \mathrm{u}_2)\cos(\theta) + (\mathrm{v}_1 - \mathrm{v}_2)\sin(\theta)]$$

$$dl = (\mathrm{u}_1 \quad \mathrm{v}_1)\begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix} - (\mathrm{u}_2 \quad \mathrm{v}_2)\begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix}$$

$$dl = \underline{d}^{i^T} \underline{\underline{T}}^i (x, y)^T \begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix} - \underline{d}^{j^T} \underline{\underline{T}}^j (x, y)^T \begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix}$$

$$dl = \underline{d}^{i^T} \underline{e} \; - \underline{d}^{j^T} \underline{g} \qquad (3.35)$$

where

$$\underline{e} = \underline{\underline{T}}^i (x_1, y_1)^T \begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix}$$

$$\underline{g} = \underline{\underline{T}}^j (x_2, y_2)^T \begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix}$$

The force in the rock bolt is

$$f = -p_b \frac{dl}{l}$$

and hence the strain energy of the bolt is

$$\Pi_b = -\frac{1}{2} f \, dl = \frac{p_b}{2l} dl^2$$

$$\Pi_b = \frac{p_b}{2l} \left(\underline{d}^{i^T} \underline{e} - \underline{d}^{j^T} \underline{g} \right)^2$$

$$\Pi_b = \frac{p_b}{2l} \left(\underline{d}^{i^T} \underline{e} \; \underline{e}^T \underline{d}^i - \underline{d}^{i^T} \underline{e} \; \underline{g}^T \underline{d}^j - \underline{d}^{j^T} \underline{g} \; \underline{e}^T \underline{d}^i + \underline{d}^{j^T} \underline{g} \; \underline{g}^T \underline{d}^j \right) . \qquad (3.36)$$

Equation 3.36 is minimised with respect to the block degrees of freedom:

$$K_{rs}^{ii} = \frac{\partial^2 \Pi_b}{\partial d_r^i . \partial d_s^i} = \frac{p_b}{2l} \frac{\partial^2}{\partial d_r^i . \partial d_s^i} \left(\underline{d}^{iT} \underline{e} \ \underline{e}^T \underline{d}^i \right) = \frac{p_b}{l} e_r e_s$$

$$\frac{p_b}{l} \underline{e} \ \underline{e}^T \Rightarrow \underline{\underline{K}}^{ii} \tag{3.37}$$

$$K_{rs}^{ij} = \frac{\partial^2 \Pi_b}{\partial d_r^i . \partial d_s^j} = \frac{p_b}{2l} \frac{\partial^2}{\partial d_r^i . \partial d_s^j} \left(-\underline{d}^{iT} \underline{e} \ \underline{g}^T \underline{d}^j \right) = -\frac{p_b}{l} e_r g_s$$

$$-\frac{p_b}{l} \underline{e} \ \underline{g}^T \Rightarrow \underline{\underline{K}}^{ij} \tag{3.38}$$

$$K_{rs}^{ji} = \frac{\partial^2 \Pi_b}{\partial d_r^j . \partial d_s^i} = \frac{p_b}{2l} \frac{\partial^2}{\partial d_r^j . \partial d_s^i} \left(-\underline{d}^{jT} \underline{g} \ \underline{e}^T \underline{d}^i \right) = -\frac{p_b}{l} g_r e_s$$

$$-\frac{p_b}{l} \underline{g} \ \underline{e}^T \Rightarrow \underline{\underline{K}}^{ji} \tag{3.39}$$

$$K_{rs}^{jj} = \frac{\partial^2 \Pi_b}{\partial d_r^j . \partial d_s^j} = \frac{p_b}{2l} \frac{\partial^2}{\partial d_r^j . \partial d_s^j} \left(\underline{d}^{jT} \underline{g} \ \underline{g}^T \underline{d}^j \right) = \frac{p_b}{l} g_r g_s$$

$$\frac{p_b}{l} \underline{g} \ \underline{g}^T \Rightarrow \underline{\underline{K}}^{jj} \tag{3.40}$$

The 6 x 6 submatrices of Equations 3.37 to 3.40 are added to the global stiffness matrix of Equation 3.5.

3.4.2 Contact submatrices

In distinct element methods, such as DDA, it is important to ensure that no block passes through another block. Since large displacements of blocks can occur in the course of an analysis, it is effectively necessary to check the location of all blocks against all other blocks periodically.

Not only must it be determined which blocks are likely to inter-penetrate in the next time-step, but the program must also make assumptions as to whether these blocks will slide against each other. In the DDA method, the system is solved, and then a check is performed to determine if the assumptions were correct. Any incorrect assumptions are revised, and the system is solved again. This process is repeated until all assumptions are found to be correct. The algorithm that controls this iterative solution method is described in Shi [1988], and it is not elaborated on here. However, when two blocks are identified as being in contact, the contact forces between them provide a component of the total potential energy of the system. The derivation and minimisation of these terms is described in this section.

Wherever blocks are in contact, a normal spring is set up working against the direction of inter-penetration.

To control sliding of blocks, Coulomb's friction law is used in DDA. This law states that, where Φ is the angle of contact friction and C is the cohesion between blocks, there is a relationship between the normal contact force, R_n, and the shear force, R_s, such that if

$$R_s \leq R_n \tan\Phi + C$$

then the blocks may not slide relative to each other. In the DDA method, a shear spring is applied. However, if

$$R_s \geq R_n \tan\Phi + C$$

then sliding can occur. A friction force acts in the direction opposite to the displacement.

The implementation of normal spring, shear spring and friction force terms is considered separately below.

For brevity, the derivation of penetration and sliding distances is not performed in this section. These derivations may be found in Shi [1988].

Normal spring

When contact between blocks is established in DDA, a small amount of inter-penetration, or overlap, of blocks is allowed. A normal spring is implemented between the two blocks, working against the direction of penetration, and the extension of this contact spring is equal to the distance of penetration. The contact spring is made very stiff to ensure that penetration distances are small compared to block dimensions.

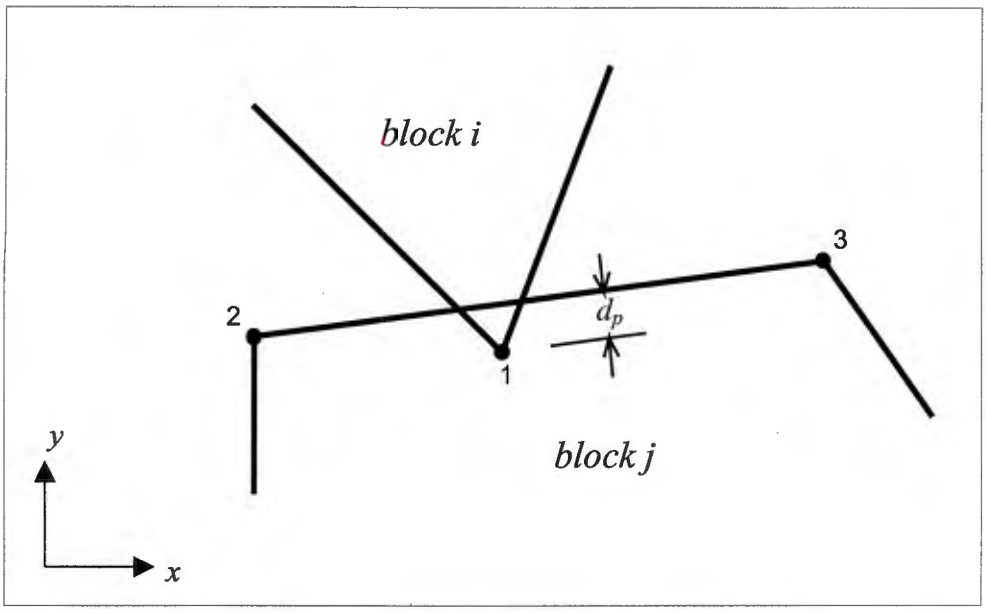


Figure 3.3: A detail of a block contact.

A detail of a contact between two blocks is shown in Figure 3.3. Point (x_1, y_1) of block i penetrates the edge between vertices (x_2, y_2) and (x_3, y_3) of block j.

It can be shown that the penetration distance d_p at the end of the time step is:

$$d_p = \frac{S}{l} + \underline{e}^T \underline{d}^i + \underline{g}^T \underline{d}^j \tag{3.41}$$

where

$$\underline{e} = \frac{1}{l} \underline{\underline{T}}^i (x_1, y_1)^T \begin{pmatrix} y_2 - y_3 \\ x_3 - x_2 \end{pmatrix}$$

$$\underline{g} = \frac{1}{l} \underline{\underline{T}}^j (x_2, y_2)^T \begin{pmatrix} y_3 - y_1 \\ x_1 - x_3 \end{pmatrix} + \frac{1}{l} \underline{\underline{T}}^j (x_3, y_3)^T \begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \end{pmatrix}$$

$$S = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix}$$

and l is the length of the edge 2-3,

$$l = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2} \quad .$$

Define

$$\Pi_p = \frac{p_c}{2} d_p{}^2 \tag{3.42}$$

where p_c is the stiffness of the normal spring, a large positive number.

$$\Pi_p = \frac{p_c}{2}\left(\frac{S}{l} + \underline{e}^T \underline{d}^i + \underline{g}^T \underline{d}^j\right)^2$$

$$\Pi_p = \frac{p_c}{2}\left(\frac{S^2}{l^2} + \underline{d}^{iT}\,\underline{e}\;\underline{e}^T\underline{d}^i + \underline{d}^{jT}\,\underline{g}\;\underline{g}^T\underline{d}^j + \underline{d}^{iT}\,\underline{e}\;\underline{g}^T\underline{d}^j + \underline{d}^{jT}\,\underline{g}\;\underline{e}^T\underline{d}^j + \frac{2S}{l}\underline{e}^T\underline{d}^i + \frac{2S}{l}\underline{g}^T\underline{d}^j\right) \quad . \tag{3.43}$$

The Total Potential Energy is now minimised with respect to the deformation variables, in the same manner as for the rock bolts:

$$K_{rs}^{ii} = \frac{\partial^2 \Pi_p}{\partial d_r^i . \partial d_s^i} = \frac{p_c}{2}\frac{\partial^2}{\partial d_r^i . \partial d_s^i}\left(\underline{d}^{iT}\,\underline{e}\;\underline{e}^T\underline{d}^i\right) = p_c e_r e_s$$

$$p_c \underline{e}\;\underline{e}^T \Rightarrow \underline{\underline{K}}^{ii} \tag{3.44}$$

$$K_{rs}^{ij} = \frac{\partial^2 \Pi_p}{\partial d_r^i . \partial d_s^j} = \frac{p_c}{2}\frac{\partial^2}{\partial d_r^i . \partial d_s^j}\left(\underline{d}^{iT}\,\underline{e}\;\underline{g}^T\underline{d}^j\right) = p_c e_r g_s$$

$$p_c \underline{e}\;\underline{g}^T \Rightarrow \underline{\underline{K}}^{ij} \tag{3.45}$$

$$K_{rs}^{ji} = \frac{\partial^2 \Pi_p}{\partial d_r^j . \partial d_s^i} = \frac{p_c}{2}\frac{\partial^2}{\partial d_r^j . \partial d_s^i}\left(\underline{d}^{jT}\,\underline{g}\;\underline{e}^T\underline{d}^i\right) = p_c g_r e_s$$

$$p_c \underline{g}\;\underline{e}^T \Rightarrow \underline{\underline{K}}^{ji} \tag{3.46}$$

$$K_{rs}^{ij} = \frac{\partial^2 \Pi_p}{\partial d_r^j . \partial d_s^j} = \frac{p_c}{2} \frac{\partial^2}{\partial d_r^j . \partial d_s^j} \left(\underline{d}^{jT} \underline{g} \ \underline{g}^T \underline{d}^j \right) = p_c g_r g_s$$

$$p_c \underline{g} \ \underline{g}^T \Rightarrow \underline{\underline{K}}^{ij} \tag{3.47}$$

$$f_r^i = -\frac{\partial}{\partial d_r^i} \left(\frac{p_c S}{l} \underline{e}^T \underline{d}^i \right) = -\frac{p_c S}{l} e_r$$

$$-\frac{p_c S}{l} \underline{e} \Rightarrow \underline{f}^i \tag{3.48}$$

$$f_r^j = -\frac{\partial}{\partial d_r^j} \left(\frac{p_c S}{l} \underline{g}^T \underline{d}^j \right) = -\frac{p_c S}{l} g_r$$

$$-\frac{p_c S}{l} \underline{g} \Rightarrow \underline{f}^j \tag{3.49}$$

The terms of Equations 3.44 to 3.49 are added to the submatrices of the global solution equations (Equation 3.5).

Shear spring

Where the shearing force between blocks is not sufficient to overcome friction, a shear lock is introduced. This is another spring which acts perpendicular to the normal spring. Figure 3.4 illustrates a block contact with friction displacement. Point 0 is on edge 2-3 of block j, at the point where point 1 of block i first penetrated block j. In this instance, the shear lock imposes a penalty function to ensure that the distance d_s is small.

A term may be derived for the sliding distance d_s, assuming small displacements,

$$d_s = \frac{S}{l} + \underline{e}^T \underline{d}^i + \underline{g}^T \underline{d}^j \tag{3.50}$$

where

$$\underline{e} = \frac{1}{l}\underline{\underline{T}}^i(x_1, y_1)^T \begin{pmatrix} x_3 - x_2 \\ y_3 - y_2 \end{pmatrix}$$

$$\underline{g} = \frac{1}{l}\underline{\underline{T}}^j(x_0, y_0)^T \begin{pmatrix} x_2 - x_3 \\ y_2 - y_3 \end{pmatrix}$$

$$S = \frac{1}{l}\begin{bmatrix} x_3 - x_2 & y_3 - y_2 \end{bmatrix} \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix}$$

and l is again the length of edge 2-3 of block j.

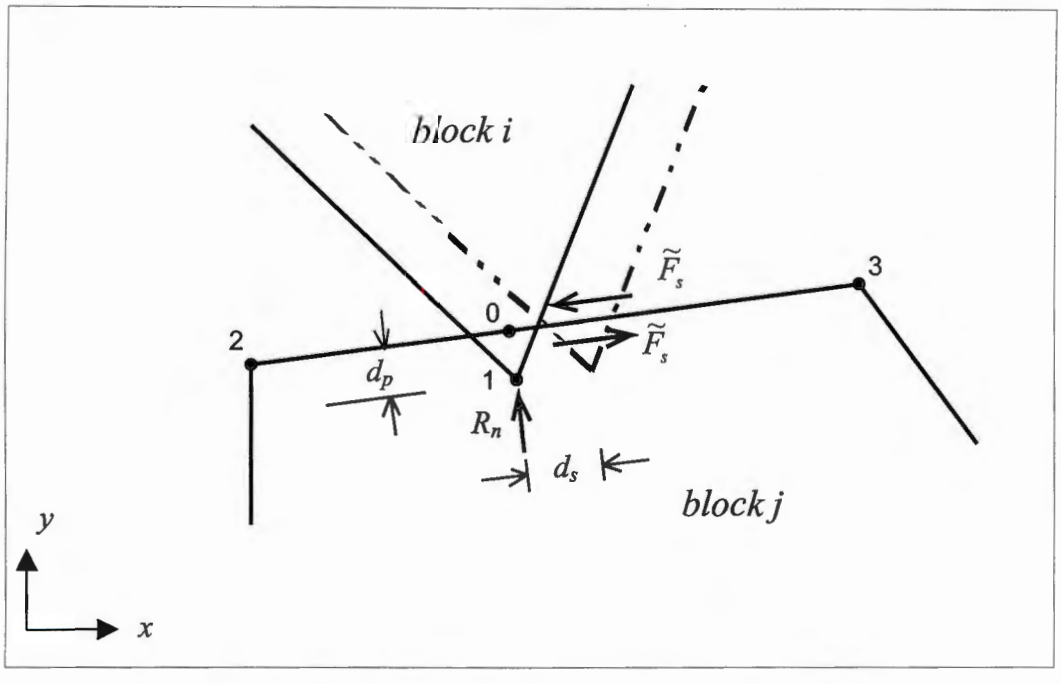


Figure 3.4: A block contact with friction displacement.

Defining the stiffness of the shear lock as p_c, the strain energy of the spring is

$$\Pi_k = \frac{p_c}{2} d_s^{\,2} \tag{3.51}$$

$$\Pi_k = \frac{p_c}{2}\left(\frac{S^2}{l^2} + \underline{d}^{iT}\underline{e}\ \underline{e}^T\underline{d}^i + \underline{d}^{jT}\underline{g}\ \underline{g}^T\underline{d}^j + \underline{d}^{iT}\underline{e}\ \underline{g}^T\underline{d}^j + \underline{d}^{jT}\underline{g}\ \underline{e}^T\underline{d}^j + \frac{2S}{l}\underline{e}^T\underline{d}^i + \frac{2S}{l}\underline{g}^T\underline{d}^j \right) . \tag{3.52}$$

Equation 3.52 is of identical form to the total potential energy equation for the normal spring (Equation 3.43), although the terms $\underline{e}$, g and S are different.

The minimisation procedure is the same as for the normal spring, and the following terms are added to the arrays of the solution equations (Equation 3.5):

$$p_c \underline{e}\ \underline{e}^T \Rightarrow \underline{\underline{K}}^{ii} \tag{3.53}$$

$$p_c \underline{e}\ \underline{g}^T \Rightarrow \underline{\underline{K}}^{ij} \tag{3.54}$$

$$p_c \underline{g}\ \underline{e}^T \Rightarrow \underline{\underline{K}}^{ji} \tag{3.55}$$

$$p_c \underline{g}\ \underline{g}^T \Rightarrow \underline{\underline{K}}^{jj} \tag{3.56}$$

$$-\frac{p_c S}{l} \underline{e} \Rightarrow \underline{f}^i \tag{3.57}$$

$$-\frac{p_c S}{l} \underline{g} \Rightarrow \underline{f}^j \tag{3.58}$$

Friction Force

The friction force, $\widetilde{F}_s$, is calculated from Coulomb's friction law:

$$\widetilde{F}_s = \{sign\} p_c d_n \tan(\Phi) + C \tag{3.59}$$

where {*sign*} is equal to 1 or –1 depending on the direction of displacement.
The energy component of the friction force is equal to the product of the magnitude of the force and the distance of sliding, d_s. Therefore, the sliding potential energy is equal to

$$\Pi_f = \widetilde{F}_s \frac{S}{l} + \widetilde{F}_s \underline{e}^T \underline{d}^i + \widetilde{F}_s \underline{g}^T \underline{d}^j \tag{3.60}$$

where $\underline{e}$, $\underline{g}$, S and l are the same as in Equation 3.50.

Minimisation of Equation 3.60 produces two load vectors

$$f_r^i = -\frac{\partial \Pi_f}{\partial d_r^i} = -\widetilde{F}_s \frac{\partial}{\partial d_r^i}\left(\underline{e}^T \underline{d}^i\right) = -\widetilde{F}_s . e_r$$

$$-\tilde{F}_s \underline{e} \Rightarrow \underline{f}^i \tag{3.61}$$

$$f_r^j = -\frac{\partial \Pi_f}{\partial d_r^j} = -\tilde{F}_s \frac{\partial}{\partial d_r^j}\left(\underline{g}^T \underline{d}^j\right) = -\tilde{F}_s . g_r$$

$$-\tilde{F}_s \underline{g} \Rightarrow \underline{f}^j \tag{3.62}$$

These load vectors are added to the global load vector of Equation 3.5.

This completes the derivation of the system solution equations.

Chapter 4

EVALUATION OF DDA VERSION 96

Discontinuous Deformation Analysis is the newest of the discrete element methods. Development of the theory and methods behind the procedure, and of the software required to solve problems, is continuing at various research centres world-wide, and at the same time the method is being applied to real problems to evaluate its suitability and accuracy. In this chapter, the performance of the DDA method is discussed.

A DDA Forward analysis is controlled by certain parameters, which are either pre-set by the user or automatically adjusted by the software during the solution process in order to optimise performance. These control parameters are introduced in Section 4.1. In Section 4.2, the method is used to solve some elementary problems, and the effect of the various parameters on the solution is investigated. Finally, in Section 4.3, some of the modifications that have been implemented by other researchers are described.

4.1 Control Parameters

The parameters that may be set by the user to optimise a DDA analysis are:

- The dynamic or static control parameter, k01. In DDA, the only difference between a static and a dynamic analysis is that in a static analysis the velocities inherited from the previous increment are reset to zero (see Section 3.3.6). If the parameter k01 is set at 1, a dynamic analysis results, while a value of 0 results in a static analysis. It is also feasible to choose a value between 0 and 1, as this will result in energy dissipation and damping in a dynamic analysis.

- The maximum displacement ratio, g2. A value for the maximum displacement ratio is required from the user. This is the ratio of the maximum incremental block displacement to half of the problem domain length. The problem domain length is a measure of the overall size of the problem, and is the greater of l_y or 1.3 l_x, as marked on Figure 2.3 of Section 2.1.1. An assumption of the DDA method is that within a single increment the displacements are small, and the accuracy of the analysis will be reduced if this is not the case. Also, if displacements are too large, then the prediction of new contacts is more complicated, and more iterations will be

required to determine the correct contact configuration. Therefore, the program documentation recommends values for g2 of between 0.001 and 0.01.

- The upper limit of the time interval, g1 (s). The DDA program controls the time interval of each increment to ensure that block incremental displacements remain within the limits imposed by the maximum displacement ratio. The user has the option of imposing a maximum time interval in order to increase the accuracy of the analysis.

- The contact penalty value, g0 (N/m). The DDA program will automatically control this value if it is not specified. Where contacts occur, stiff springs ensure that the inter-penetration of the blocks remains small. The value g0 is the contact force per unit penetration. This value affects the accuracy with which dynamic collisions are modelled. The program documentation recommends a value for g0 of the product of the Young's modulus of the blocks and the average block diameter.

4.2 Application of DDA to Elementary Problems

Four example problems are modelled in this section, using DDA Version 96. Each problem is designed to investigate a different aspect of the simulation. In all of these analyses, the Young's modulus and Poisson's ratio are 40 GPa and 0.3 respectively.

4.2.1 Dynamic sliding

This problem models a single block sliding on a rough surface. The block has an initial velocity of 10 m/s, and it comes to rest under the forces of friction, where the angle of friction, Φ, is 10°. In Figure 4.1, the displacement of the block with time is shown for various simulations, employing different upper limits of the time interval (parameter g1).

The simulations are not affected by the penalty value (g0). The maximum displacement ratio (g2) was set at 0.03.

With a time interval, Δt, not greater than 0.005 seconds, the path of the block with time is indistinguishable from the theoretical path. Thus DDA has the potential to accurately model the Coulomb friction model in dynamic analyses. However, with increasing values of g1 the runs become less accurate. If no upper limit to the time interval is set, the DDA analysis starts with a time interval of approximately 0.03 seconds, but this increases during the analysis as the block slows down. The incremental time interval is plotted against the scale on the right hand side of Figure 4.1. As a result of this time increment increase, the precision of the analysis is reduced, and the actual distance that the block travels before coming to rest differs from the theoretical value by 22%.

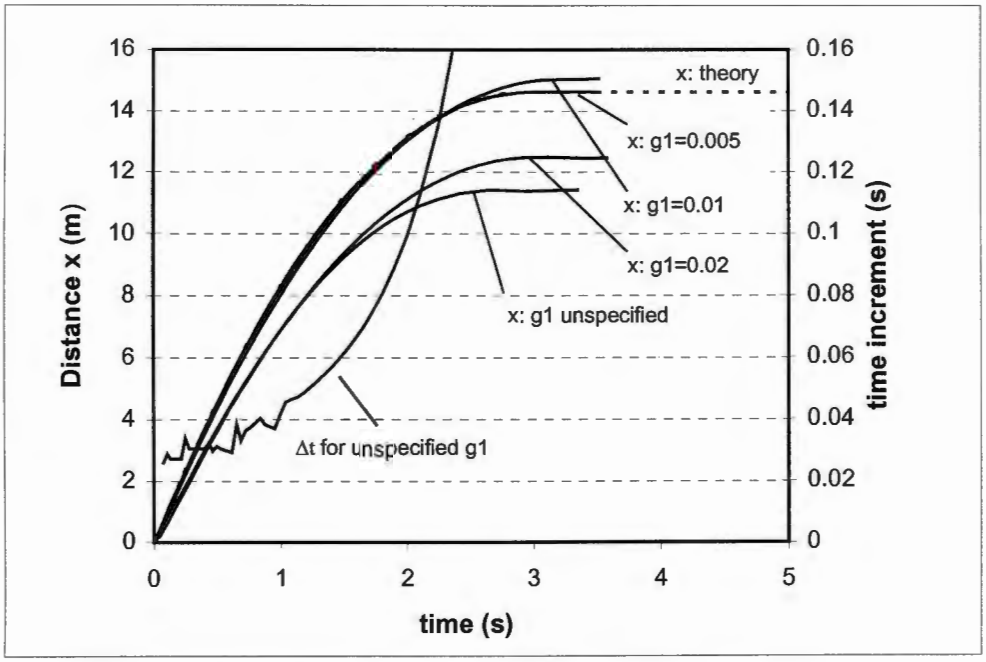


Figure 4.1: Distance-time relationship for a block sliding on a rough surface, using different values of g1.

4.2.2 Dynamic contact

A schematic showing the geometry of this problem is shown in Figure 4.2. An octagonal block, block 1, falls under gravity, and strikes the top surface of block 2. The velocity of block 1 at impact, v_i, may be controlled by altering its initial velocity. Block 2 is fixed at points A and B, so that its only degree of freedom is in vertical strain.

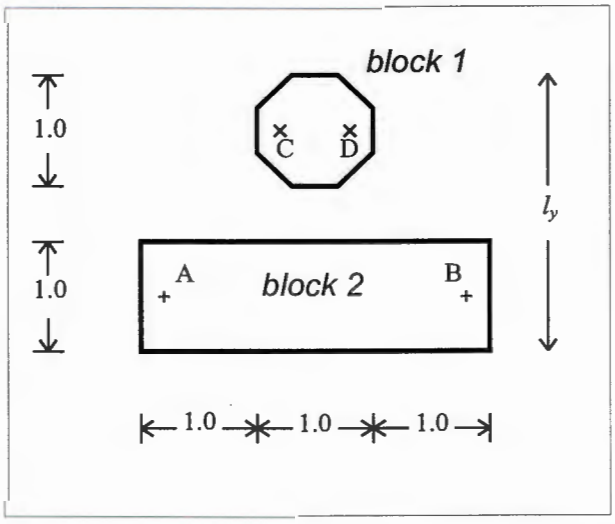


Figure 4.2: Schematic of the dynamic impact simulation.

The DDA method accurately models the motion of a block falling freely under gravity in dynamic analyses. This simulation is concerned with the behaviour of block 1 when it comes into contact with block 2. The only way to control energy damping in the DDA method is to alter the dynamic or static control parameter, k01. If this is done, however, the block will no longer accelerate as expected under gravity, because the velocity is factored at the beginning of each time increment. Therefore, no energy damping was specified for this simulation. It would be expected that the kinetic energy of the block after collision should be equal to that of the block immediately before collision, less the energy that goes into setting up dynamic vibration in both blocks as a result of the collision.

The control parameters that affect this analysis are the penalty value, g0, the maximum time interval, g1, and the maximum displacement ratio, g2. The first test series was conducted with varying values of the maximum displacement ratio, and with the contact penalty value and time interval controlled automatically by the program. The impact velocity of the block, v_i, was 5 m/s. For each run, the kinetic energy of the block after impact was calculated as a percentage of that before impact. The results of this series are shown in Figure 4.3.

No clear pattern is apparent from this test series, and the results vary widely. It is interesting to note that in most cases a large proportion of the kinetic energy is lost in the collision, despite the lack of any damping mechanism. It was not possible to determine the amount of energy absorbed in dynamic vibration of the blocks, because the time step interval is approximately the same as the period of vibration, so that vibration was not well modelled by the program.

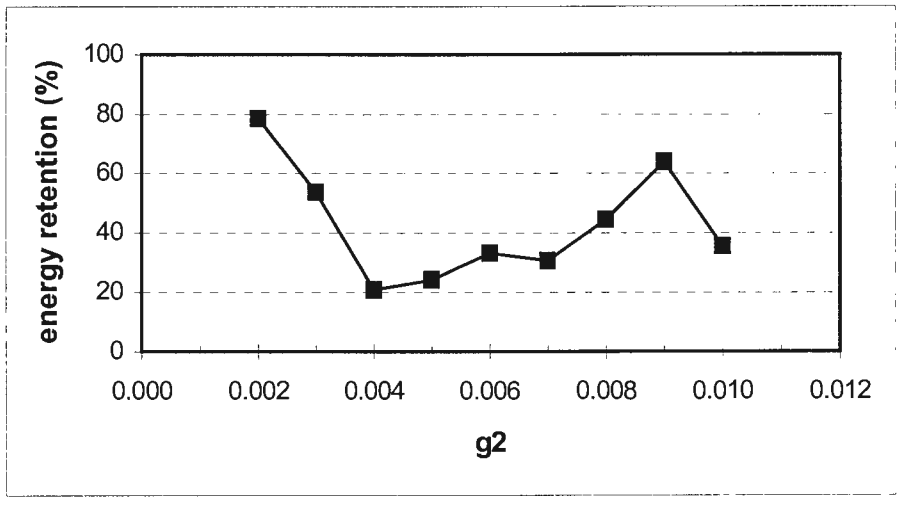


Figure 4.3: Kinetic energy retention in collision, for various values of g2.
(v_i=5 m/s; g0 and g1 not specified.)

The size of the blocks in this analysis is large in comparison to the 'half-domain' length, which is half of the dimension l_y in Figure 4.2. Therefore, for the remaining test series the value of g2 was set at 0.01, at the upper end of the recommended range.

In the next test series, the impact velocity of block 1 was varied. Parameter g2 was set at 0.01, and g0 and g1 were again not specified. Figure 4.4 shows the percentage of energy retention in collision at these velocities. It can be seen that energy retention is reduced as the impact velocity of block 1 increases. In high velocity collisions, almost all of the energy is dissipated.

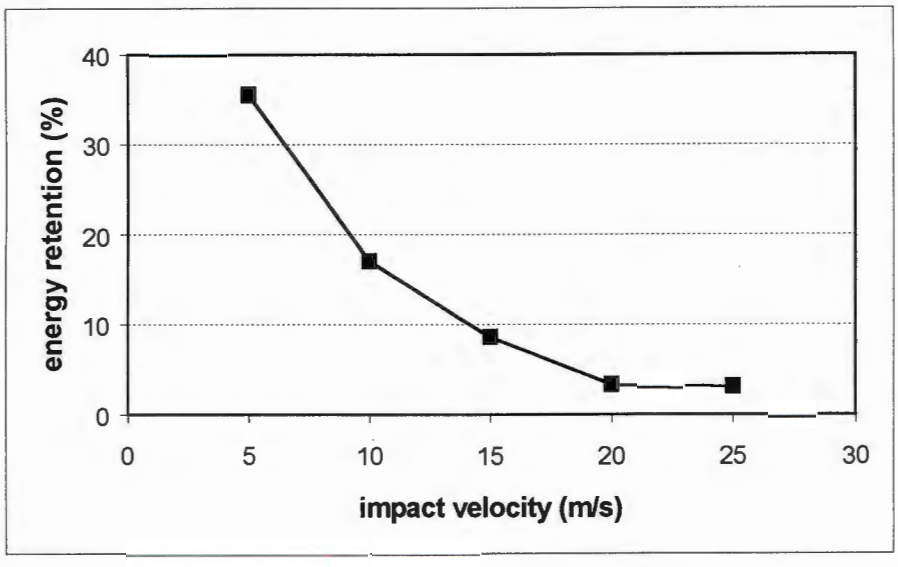


Figure 4.4: Kinetic energy retention in collision, for various values of v_i. (g2=0.01; g0 and g1 not specified.)

In order to determine why energy loss is so high, it is necessary to look at the collision simulation in more detail. For the case of the block with an impact velocity of 5m/s, the velocity of the block and the inter-penetration between the blocks is plotted against time in Figure 4.5. Penetration is measured at the mid-point of the common contact surface. It can be seen from this plot that the blocks are only in contact at the end of a single time increment. This appears to be insufficient to accurately model the collision.

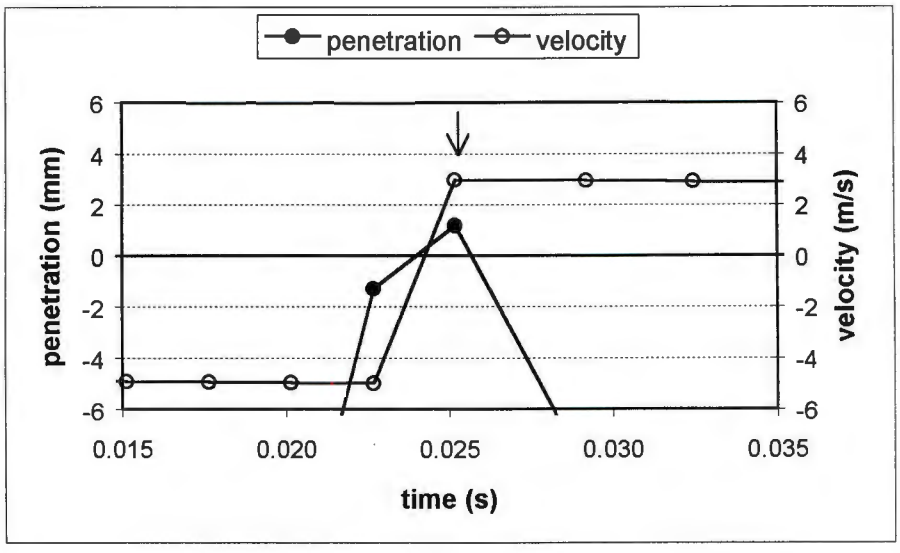


Figure 4.5: Penetration and velocity against time, for a block moving at 5 m/s at impact.

The time-step and penalty values are controlled by the program in this analysis. These values are plotted in Figure 4.6. The arrow marks the point at which penetration occurs.

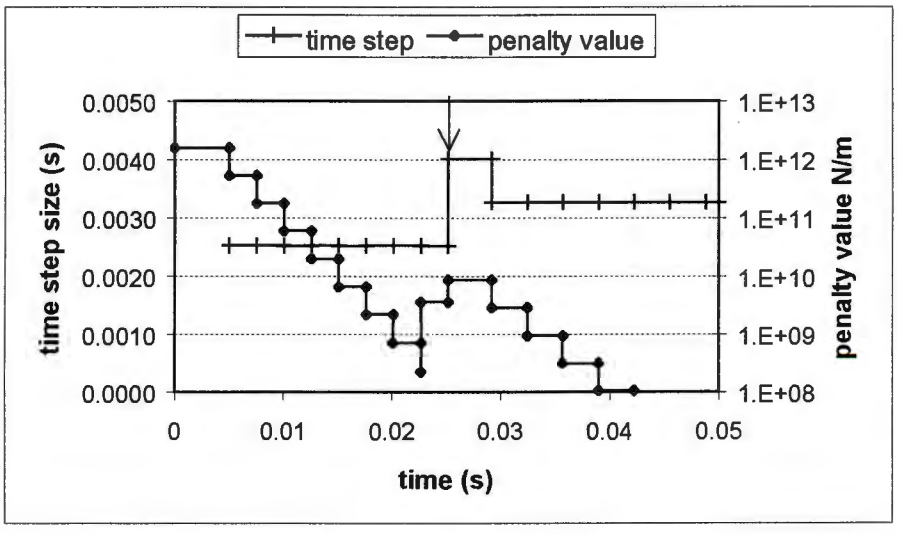


Figure 4.6: Time-step and penalty values against time, for a block moving at 5 m/s at impact.

If the penalty value is not specified, the program automatically sets it at an initial value of 40 times the average Young's modulus of the blocks. In this case, this value is 1.6 x 10^{12} N/m. However, there is no contact in the initial stages of the run, and so the value is automatically reduced by a factor of 0.3 in every subsequent time-step. When contact does occur, the penalty value has therefore reduced to 2.44 x 10^{8} N/m. This value is insufficient to contain penetration to within the allowable value, so the program increases the penalty value to 3.57 x 10^{9} N/m and attempts a second iteration. This

time, the penetration is within the limit, and the program proceeds to the next increment. The penalty value is recalculated as 8.60 x 10^9 N/m. However, the blocks already have sufficient relative velocity to re-exit within this increment, so that the contact spring is not implemented. In subsequent increments, there are no contacts, and the penalty value is reduced by a factor of 0.3 for each increment, as before.

The time-step interval is calculated on the basis of incremental displacements, which are in turn dependent on the velocity of the block. Before contact, the time interval is constant at a value of 0.252s. In the increment where contact occurs, the average velocity of the block is reduced, and the time interval increases. As the block exits, the time step interval is reduced again to reflect the new velocity of the block, but because the block is now moving slower, the time interval is greater than that before impact.

It is clear that dynamic collisions cannot be reliably modelled with such a coarse time step interval. In the next series, the displacement ratio, g2, remained at 0.01, and the impact velocity at 5 m/s. The penalty value was not specified, but the incremental time interval was reduced to various values by specifying parameter g1. Figure 4.7 shows kinetic energy retention for the various specified time intervals, where the time interval is plotted in milliseconds.

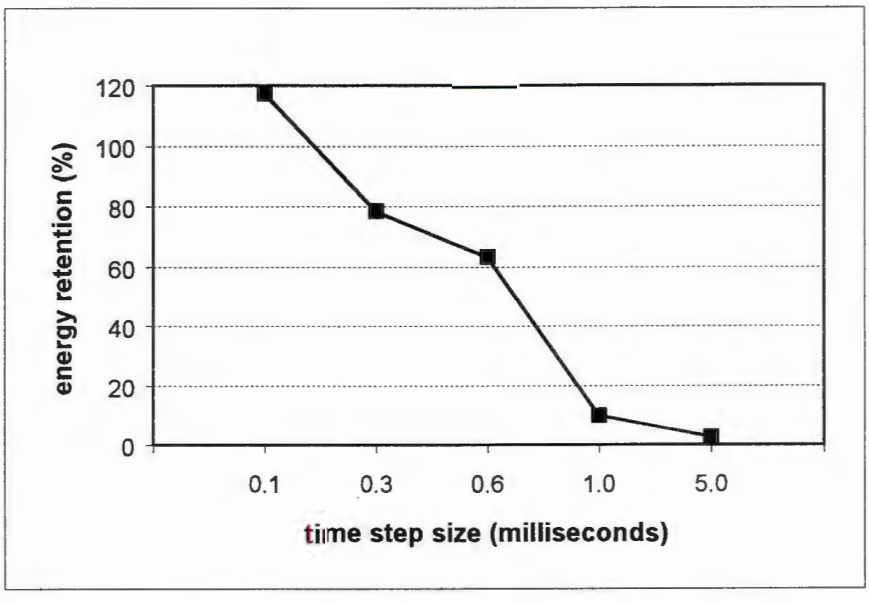


Figure 4.7: Kinetic energy retention in collision, for various values of g1.
(v_i=5 m/s; g2=0.01; g0 not specified.)

It can be seen that as the time step interval is increased, the energy retention decreases. However, for a time interval of 1 x10^{-4}s, the kinetic energy of the block actually increases to a value greater than that before the collision. In Figures 4.8 and 4.9, this collision is depicted in more detail.

Figure 4.8 shows the block velocity and penetration distance for a portion of the analysis. In Figure 4.9, the value of the penalty value is plotted against time for the same time period.

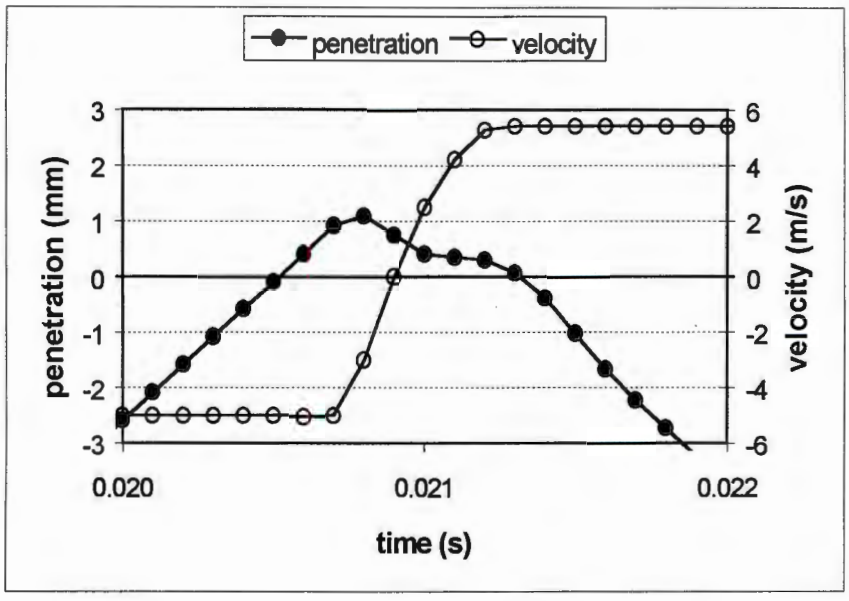


Figure 4.8: Velocity and penetration for g1=0.0001.
(v_i=5 m/s; g2=0.01; g0 not specified.)

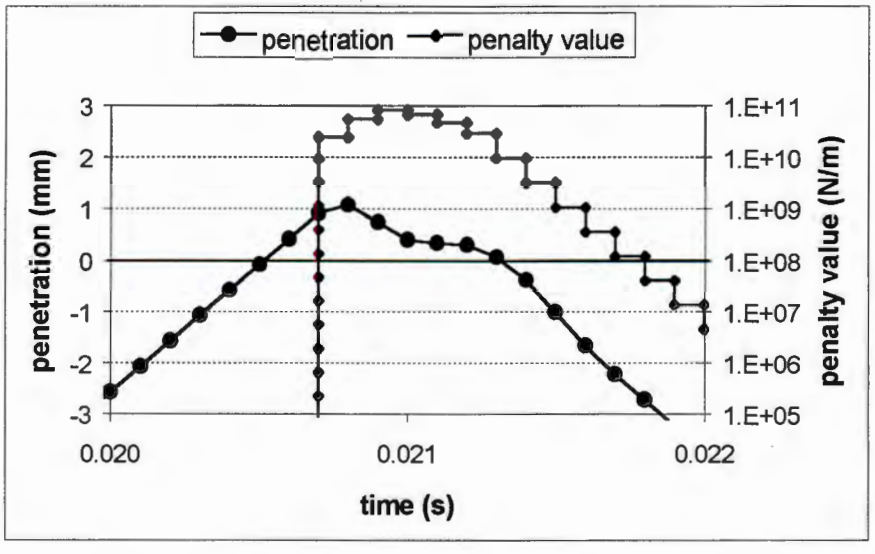


Figure 4.9: Penalty value variation during the analysis.
(v_i=5 m/s; g2=0.01; g1=0.0001; g0 not specified.)

Because the time-step interval has been reduced, contact only occurs in the 205th increment of this analysis. The initial penalty value was determined by the program to be 1.6 x 10^{12} N/m as before, but this progressively reduces while there is no contact, and is effectively zero when impact occurs. For the first two increments after contact,

the penetration spring is implemented, but because the penalty value is so low, the velocity of the block is unaffected.

However, the DDA program monitors penetration, and does not allow the value to increase beyond a ratio of 0.001 of the domain half-length. In increment 207 of the analysis, the penetration exceed this ratio, and so the DDA program tries further iterations, increasing the penalty value by a factor of 3.0 on each attempt. On the 37^{th} iteration, the penalty value has increased to a level sufficient to limit the penetration, and the analysis continues. Once again, the penalty value is recalculated for each increment, and at first it increases. When the contact opens, the penalty value progressively decreases, as before.

An anomaly appears to occur when block 1 is re-exiting block 2. Although the block is accelerating, the penetration distance remains almost constant for three increments. This is because the blocks are both elastic, and, having been initially compressed by the impact, they are rebounding. This extends the period in which the block and the surface are in contact, and is possibly the reason why the kinetic energy of the system increases.

The control of the penalty value by the DDA program is not satisfactory. As a final step in modelling dynamic contact, therefore, a realistic contact value has to be specified.

The next test series was conducted with a range of different values for the penalty value. The time step interval was specified as 1×10^{-4}s, with the other values remaining the same as in the previous series.

If the penalty value is too low, inter-block penetration cannot be maintained within the allowable limit, and the program will not complete the run. In the results shown in Figure 4.10, the first value of the penalty value is close to the minimum value. This value of 2.0×10^{10} is comparable to the recommended value of the product of the Young's modulus and the average block diameter. In this analysis, this recommended value would equal 6.0×10^{10}. The kinetic energy retention is plotted against the right hand axis, and the number of increments for which contact is maintained is plotted in column format against the left hand axis.

As the stiffness of the contact spring is increased, the contact period reduces, and the energy loss increases. In Figure 4.11, a detail of the penetration and velocity of the contact is plotted. It can be seen that, although there is still some energy loss in this simulation, the contact is reasonably accurately modelled.

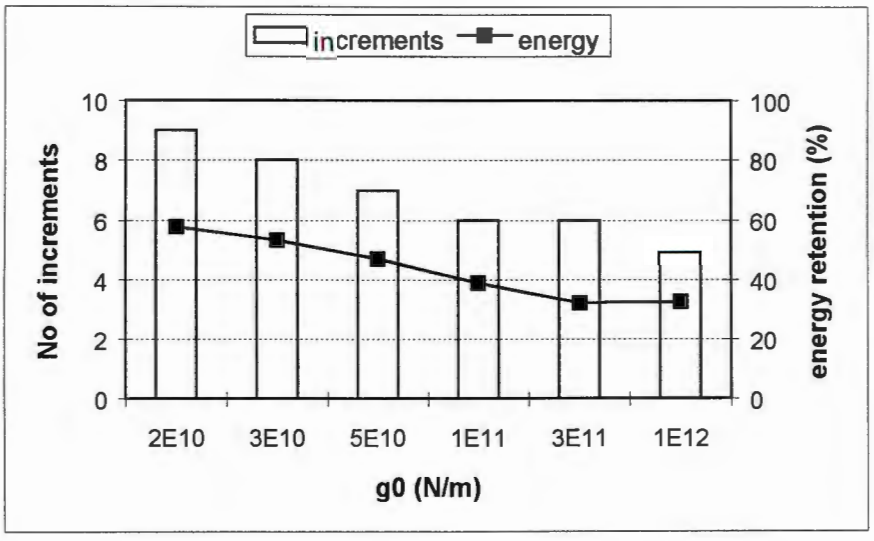


Figure 4.10: Kinetic energy retention in collision, No. of contact increments for various values of g0. (v_i=5 m/s; g1=0.0001; g2=0.01.)

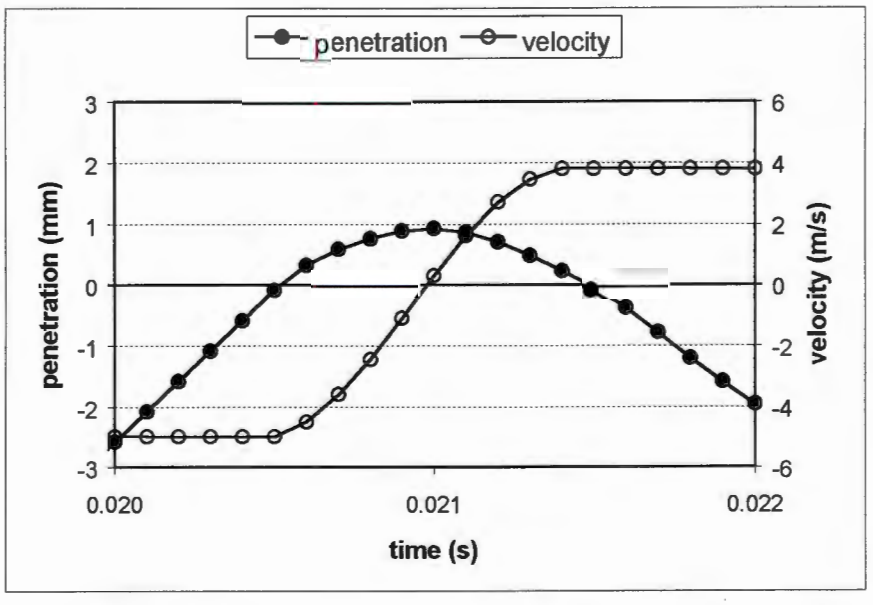


Figure 4.11: A detail of velocity and penetration during contact for g0=2.0 x 10^{10} N/m.

The final simulation shown is a reasonably accurate simulation of contact between the two blocks. More than 40% of the kinetic energy is still lost in the collision. In actual collisions between rock blocks, a high energy loss is to be expected, and so it is fortuitous that energy is lost in DDA collisions. However, the method is not consistent in modelling collisions. If the accuracy of the simulation is increased by reducing the time-step period, this also increases the computational cost of the analysis.

Because of the wide divergence of energy loss, it appears that the DDA method is not suitable for modelling dynamic collisions, and it should be used with caution in such applications as rock-fall simulations, where block velocities can become large.

4.2.3 Static contact

In this test series, static analyses were conducted. The simulations had a similar geometry to those for the dynamic contact studies, except that block 1 was resting on block 2. Two equal point loads were applied vertically downward to block 1 at points C and D of Figure 4.2. The time increment is artificial in static simulations, as inertia is not considered, and the system is solved for static equilibrium for every increment. A time step of one second was therefore selected, with the load increasing linearly for the first 80 seconds, and then remaining constant for the remainder of the run.

Two analyses were conducted. In the first, the penalty value was specified, whereas in the second it was controlled by the program. Figure 4.12 shows the results of the analyses. It can be seen that the penetration characteristics are very different in the two cases. Where the penalty value is specified, penetration is linearly dependent on the load, while if the penalty value is not specified, then the DDA program varies the penalty value in order to obtain an approximately constant penetration.

The loading causes vertical strains in both of the blocks. These strains are also plotted in Figure 4.12 against the right hand scale. However, for the two analyses, the strain-time curves for each block are indistinguishable.

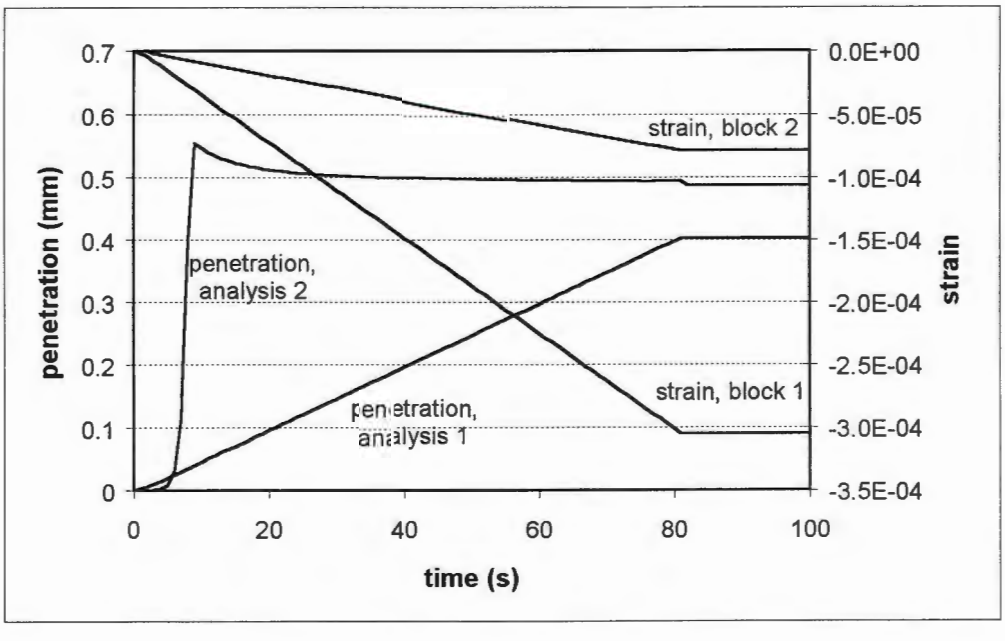


Figure 4.12: Penetration and strains between blocks in contact, under normal load.

In this analysis, it is to be expected that the applied loads would be transferred through block 1 into block 2. As a check on this, a third analysis was run, with the loads applied to block 2, at the points where the corners of block 1 were previously in contact with it. In Figure 4.13, the displacement of the top surface of block 2 is compared with that of the previous analysis with automatic control of the penalty function, and again, the strains in the lower block are compared. The two analyses yield very similar results.

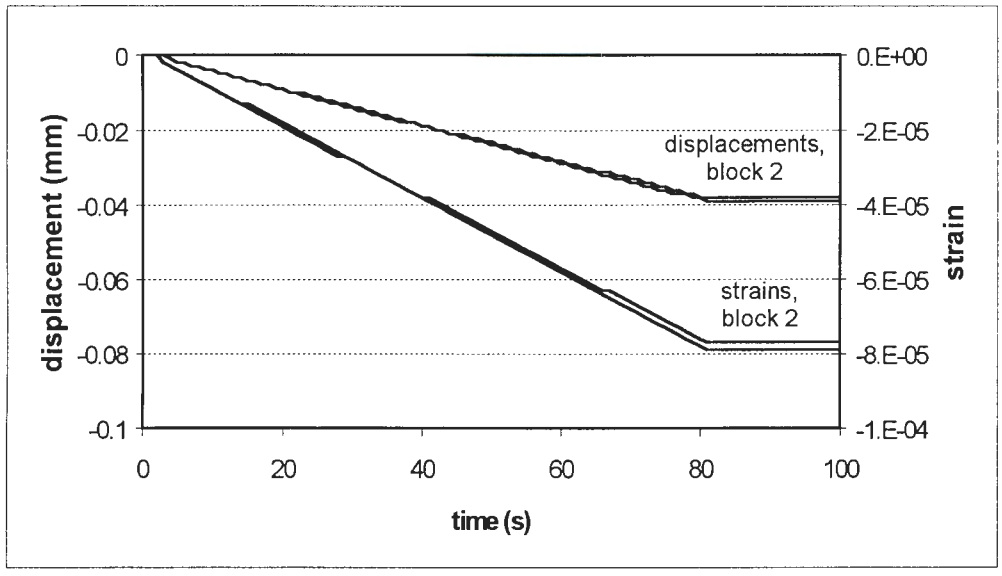


Figure 4.13: Displacement of the top surface and strain in block 2 against time.

The modelling of soft contacts is artificial in the sense that in reality bodies do not overlap. What happens, in fact, is that local deformation of the material occurs in the region of the contact, and so the geometric effect is similar. The DDA method does therefore appear to model static contacts effectively, and is suitable for simulation of rock masses, where contact between blocks is wide-spread, static, and in general maintained by normal compressive stresses due to the self-weight of the material.

4.2.4 Stress determination

The first order displacement function used by DDA is not sufficiently accurate to model the displaced shapes of blocks under complex loading. However, it does make it possible to obtain general values for the stresses in a rock block, which would not be possible if the blocks were assumed to be rigid. To test the accuracy of this stress determination, a simulation was performed of a rock mass loaded by a foundation footing resting at ground level. In order to simulate an elastic continuum, the frictional values of the joints in the rock mass were made high (C=500 kPa, Φ=50°) to minimise sliding of the blocks relative to each other, and the ends of the blocks were also bevelled in order to lock the blocks together. The outer boundary of the rock mass was

made frictionless, however. The geometry of the system is shown in Figure 4.14. The simulation was run as a static analysis.

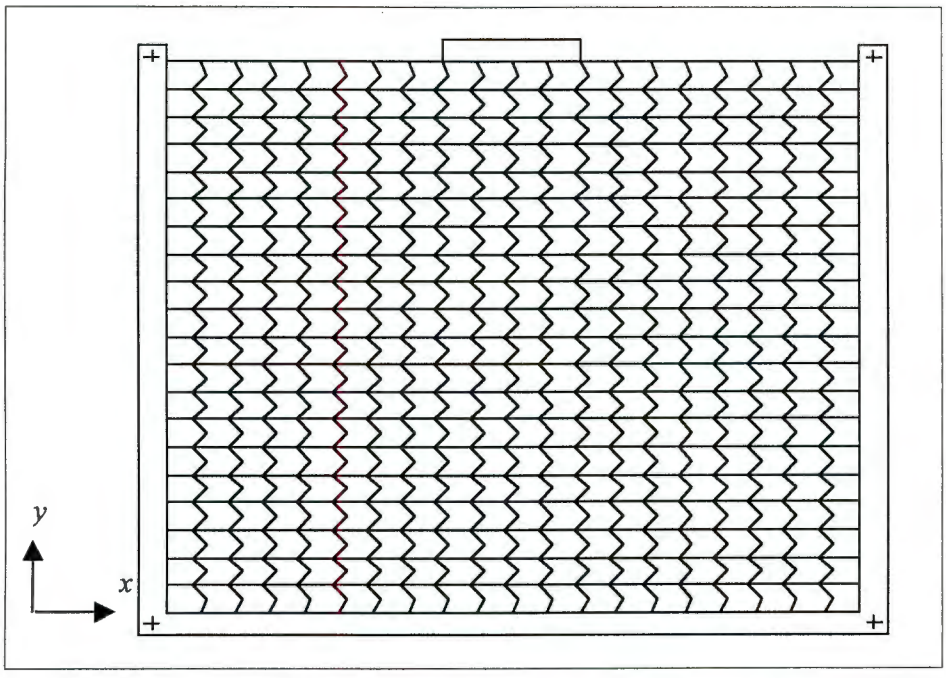


Figure 4.14: Problem geometry for a rock mass below a footing.

A theoretical solution for the stresses below a footing can be obtained from elastic theory [Craig, 1992], and a contour plot of the theoretical stress in the y-direction is shown in Figure 4.15. The solution obtained using the DDA method is shown in Figure 4.16.

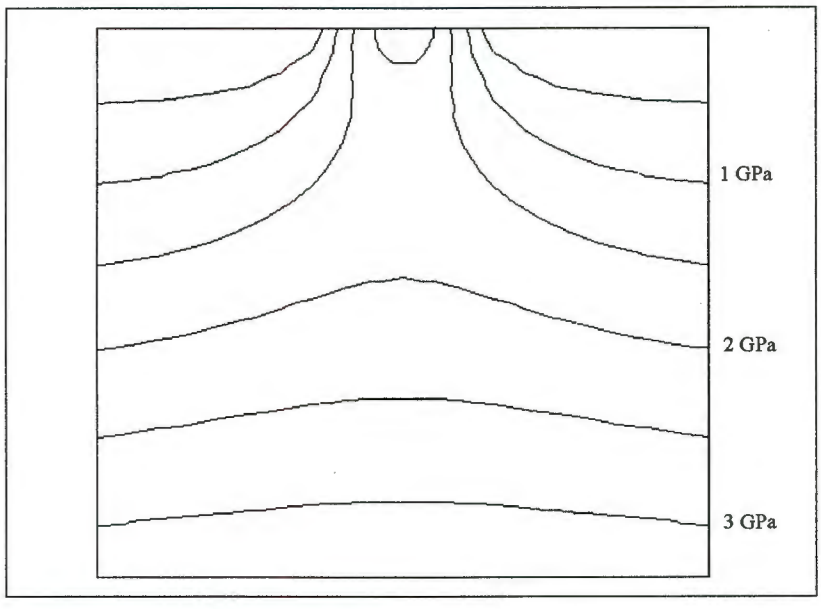


Figure 4.15: Theoretical stress σ_y in a rock mass below a footing.

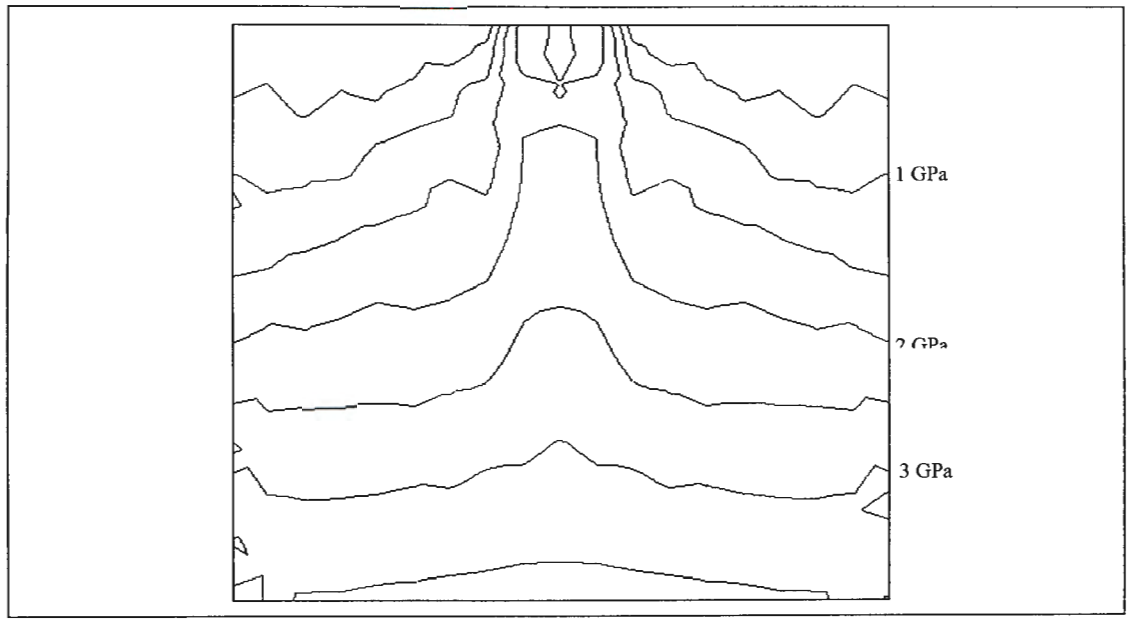


Figure 4.16: DDA solution for stress σ_y in a rock mass below a footing.

The stresses obtained in the DDA analysis are in reasonable agreement with the theoretical solution, although the stress distribution is not smooth. It should be noted that the irregularity of the stress distribution increases markedly if the blocks are of random, irregular shapes.

4.2.5 Recommendations

The results of the tests set out in this section indicate that the values specified for the control parameters do have an effect on the quality of the solution obtained. A user should be aware of this, and should understand the implications of each control parameter. Further, the user should be prepared to conduct a series of runs so as to optimise the analysis.

Any analysis is a trade-off between computational accuracy and computational cost, so that it is difficult to make general recommendations as to what parameters should be specified. However, some points do emerge from the tests set out above:

- DDA should not be used where block velocities are large. Other analysis methods are more suitable in these situations, in particular momentum exchange methods [Hahn, 1988].

- Because the incremental time step is calculated based upon block velocities, time increments can become very large if block velocities are small. This may reduce

the accuracy of the analysis, and it is recommended that a value for g1 is specified so that time increments do not become excessively large.

- The penalty value is generally controlled more on the basis of mathematical convenience than on the simulation of actual contacts in the DDA method. In the analysis depicted in Figure 4.9, the penalty value reduces from 1.6 x 10^{12} N/m to 1.0 x 10^{-6} N/m in less than 20 milliseconds. It would be practical to alter the program so that the penalty value can not fall below a certain, minimum, level. Failing this, it is recommended that the user specify a fixed penalty value. The product of the Young's modulus and the average block diameter is suitable for this. An alternative method for determining an appropriate penalty value is to carry out an analysis without specifying a penalty value, and then to find the maximum value that the penalty value reaches in the course of the analysis, using this as a basis for further analyses.

4.3 Extensions and Modifications to the DDA Method

The DDA software package discussed in this chapter represents the original concept developed by Shi [1988]. Various modifications have been proposed for the DDA method. In this section some of the more significant modifications by various researchers are described.

4.3.1 Modelling of contacts

In order to improve the efficiency of contact solving, a Lagrangian method may be used to determine the contact force between blocks [Lin, 1995]. In this method, the contact force becomes a further degree of freedom, which is solved for simultaneously with the block unknowns. This makes it unnecessary to determine contact stiffnesses by iterative solution.

This method increases the number of degrees of freedom of the system, and hence the computational cost. An alternative approach is described by Amadei *et al.* [1996], in which an augmented Lagrangian method is used with iterative solving. In this method, the strain energy of the contact spring is described by the equation

$$\Pi_p = \lambda_k^* d + \frac{1}{2} p_c d^2 \qquad (4.1)$$

where λ_k^* is a pre-determined Lagrange multiplier, and p_c is the penalty value as before.

The penalty value is now solved for iteratively, to satisfy penetration limits. For the next increment, the Lagrange multiplier is updated by the equation

$$\lambda^*_{k+1} = \lambda^*_k + p_c d \quad . \tag{4.2}$$

The introduction of the Lagrange multiplier reduces the dependence of the solution on the penalty value, and so the number of increments necessary to determine the correct value for the penalty value is reduced, while the number of system equations remains the same.

4.3.2 Rotational dilatation

When a block undergoes rigid body rotation in DDA, it tends to increase in size. The displaced position of a point (x, y) in a rotating block is given by

$$\begin{aligned} u &= -\bar{y}\ r_0 \\ v &= \bar{x}\ r_0 \end{aligned} \tag{4.3}$$

where r_0 is the incremental rotation of the block, in radians.

This first order approximation is satisfactory only if incremental rotations are small. If rotations are too large, errors are introduced [Yeung, 1991]. This situation is illustrated in Figure 4.17.

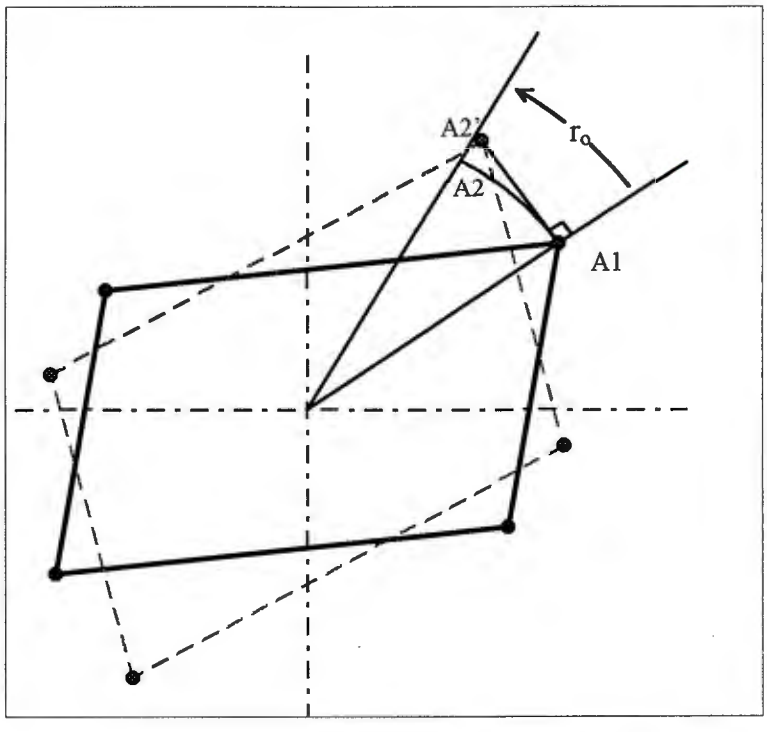


Figure 4.17: A block under rigid body rotation.

If the block of Figure 4.17 rotates by an angle of r_0, then point A1 should move to position A2. Because a first order displacement function is used, it actually moves to position A2'. This error applies to all the boundary nodes, and the cumulative effect over many increments can lead to considerable increases in block size.

A simple method of avoiding dilation is described by Ke [1996]. The first order displacement function is used for the equilibrium calculations, but when calculating the displaced position of the block, an augmented function is used:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \bar{x}(\cos(r_0) - 1) - \bar{y}(\sin(r_0)) \\ \bar{x}(\sin(r_0)) + \bar{y}(\cos(r_0) - 1) \end{pmatrix} + \underline{\underline{T}}^i(x, y)\underline{d}^i \quad . \tag{4.4}$$

This method eliminates the dilation effect. Because the energy calculations are based on the first order displacement function, incremental rotations should still be limited to less than 0.1 radians.

4.3.3 Stress Determination

The DDA method does not allow more than a very general determination of the stresses in a rock mass. It is sometimes desirable to obtain a more detailed description of the stresses in blocks within the system. Three approaches have been taken to allow this, and these are described separately below.

Higher order blocks.

Blocks can be created that are of a higher order than the first-order blocks found in the basic DDA code. Second-, and even third-order blocks have been incorporated into the original DDA code [Chern *et al.*, 1995; Koo and Chern, 1996]. With a greater number of degrees of freedom, the strains in a block may vary in a linear or quadratic manner across its area.

In both of the formulations mentioned above, a general form of polynomial displacement function was used. For the second order block, this is

$$u = a_o + a_1\bar{x} + a_2\bar{y} + a_3\bar{x}^2 + a_4\bar{x}\bar{y} + a_5\bar{y}^2$$
$$v = b_0 + b_1\bar{x} + b_2\bar{y} + b_3\bar{x}^2 + b_4\bar{x}\bar{y} + b_5\bar{y}^2$$

where $a_0,.., a_5$ and $b_0,.., b_5$ are the twelve degrees of freedom of the block.

The third order block includes the third order terms and has a similar displacement function, with twenty degrees of freedom.

The formulation of the energy equations for these blocks is along the same principles as for the first order blocks, and the second order block is discussed further in Chapter 6.

Sub-blocking

The 'artificial joint concept' of Ke and Goodman [1994] divides a block into a number of triangular sub-blocks. The result is similar in appearance to a Finite Element mesh of triangular elements within the block. Each of the sub-blocks is formulated as a normal DDA element, and then neighbouring sub-blocks are connected by penalty functions to ensure that they remain locked together.

The penalty functions have the same formulation as the normal and shear springs used in contacts between blocks in the standard DDA method. Two points are chosen along the common boundary between sub-blocks. Each of these points lies on both blocks, and the two blocks are connected by two orthogonal stiff springs at each point. The springs initially have zero length. If the blocks displace relative to each other, then the springs extend, and this increases the potential energy of the system as a whole.

Ke and Goodman [1994] used the augmented Lagrangian method to model these springs.

Mixed formulations

The DDA method may be combined with the Finite Element Method. Each block can then be sub-meshed and solved as a finite element mesh, while the interaction of the blocks is controlled using DDA. Chang [1994] has developed a program in which the potential energy of the DDA blocks and the FEM sub-meshes is minimised simultaneously, producing an overall equilibrium solution.

4.3.4 Fracture

A fracture mechanism may be conveniently built into the sub-blocking method of Ke and Goodman [1994]. The forces in the sub-block connection springs are tested against a failure criterion, such as the Mohr-Coulomb criterion. If the force in a spring exceeds the failure limit, the spring is released, and separation or sliding may occur along the sub-block boundary.

4.3.5 Material non-linearity

The only context in which non-linear materials have been modelled in DDA is in the mixed formulation of Chang [1994]. The Finite Element Method is well suited to incorporating non-linearity, and Chang was able to introduce elastic-plastic behaviour into the local Finite Element meshes of his method.

Chapter 5

MESHING IN DDA USING 1ST ORDER ELEMENTS

5.1 Introduction

A numerical modelling program must make certain simplifications in order to arrive at an approximate solution to a real problem, which effectively has an infinite number of degrees of freedom.

One important simplification that the DDA method makes is that of constant strain within each block. This assumption is reasonable where relative movement along discontinuities is the primary factor in deformation and failure of the rock mass, rather than stresses and strains in the intact material. However, it is often desirable to make a more accurate determination of the stresses in the blocks, or in one or more critical blocks.

This is particularly the case in engineering applications where it is often necessary that a structural element interact with the rock mass. Examples of such applications are concrete tunnel linings, supports in mine excavations, rockfall barriers at road cuttings, etc. In designing these structures, some form of assessment of the stresses that are likely to develop is necessary.

Several approaches have been suggested and implemented to allow this, and these were described in Section 4.3.3.

The use of higher order elements, as implemented by Chern *et al.* [1995] and Koo and Chern [1996] does allow for a more accurate determination of strains, and thence stresses. However, the strains across the block are only allowed to vary in the specified manner. Localised stress concentrations, in particular, are not well modelled. The advantage of using a sub-meshing method is that the accuracy of the strain determination can be controlled by adjusting the refinement of the mesh, either locally or as a whole.

A new method of sub-meshing is proposed here. The method results in a Finite Element mesh within blocks, but the stiffness matrices and load vectors for the sub-block elements are determined in an identical manner to those for other, single-element blocks, using the standard DDA code. Only where the DDA code is required to assemble these separate matrices into a global stiffness matrix does the meshing

method diverge from the DDA method. The sub-block arrays are converted by means of a transformation matrix, so as to solve for nodal displacements, rather than the rigid body motions and strains. These transformed arrays are incorporated into the DDA global solution, and the stiffness matrix is solved. The nodal displacements are then transformed back to the sub-block deformation parameters. Arrays for blocks that are not sub-meshed are not transformed, and are solved in their normal DDA form.

This procedure is considerably simpler than the mixed formulation developed by Chang [1994]. While it bears a closer similarity to the artificial joint method of Ke and Goodman [1994], it does have one important advantage. The number of degrees of freedom of a sub-meshed block is fewer than that of all of the independent sub-blocks. As an example of this, consider the simple mesh in Figure 5.2 of Section 5.4. This mesh may be modelled in the Finite Element Method using first order triangular elements, and each of the five nodes has two degrees of freedom, resulting in 10 degrees of freedom in all. In Ke and Goodman's approach [1994], each of the four sub-blocks retains its six degrees of freedom, a total of 24. The extra degrees of freedom are restrained by the penalty functions, and do not contribute to the accuracy of the solution. The proposed method eliminates these superfluous degrees of freedom, producing a smaller, yet still symmetric, global stiffness matrix, with resultant savings in solution time.

The theory behind the method is set out in sections 5.2 to 5.5.

5.2 Displacement Variables

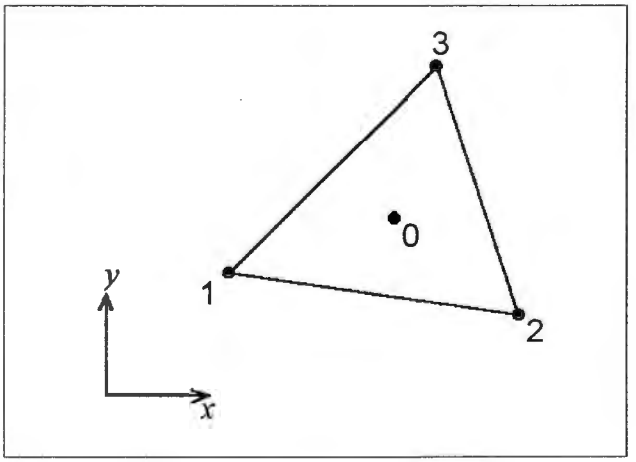


Figure 5.1: A single triangular block.

Consider the triangular block shown in Figure 5.1. In the DDA method, each block has six degrees of freedom. These are the displacements of the centroid (point 0 in Figure 5.1), u_0, v_0, and r_0 and the strains ε_x, ε_y and γ_{xy}, which are assumed to be constant across the block. The displacement of any point in the block can be determined if the

values of the six deformation parameters are known. Because the block edges remain straight, an equivalent description of the displaced shape of the block can be obtained by determining the displacements of the corner nodes. This will again result in six unknowns: (u_1, v_1), (u_2, v_2) and (u_3, v_3). In Section 3.1, it was shown that the displacement of a point in a block can be obtained by the equation:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{\bar{y}}{2} \\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{\bar{x}}{2} \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \\ r_0 \\ \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}$$

$$\begin{pmatrix} u \\ v \end{pmatrix} = \underline{\underline{T}}^i(x,y)\underline{d}^i \qquad (5.1)$$

If the undisplaced co-ordinates of the block corner nodes are substituted into $\underline{\underline{T}}^i(x,y)$, then an a transformation matrix giving the nodal displacements in terms of the deformation parameters may easily be developed.

$$\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \end{pmatrix} = \begin{bmatrix} \left[\underline{\underline{T}}^i(x_1,y_1)\right] \\ \hline \left[\underline{\underline{T}}^i(x_2,y_2)\right] \\ \hline \left[\underline{\underline{T}}^i(x_3,y_3)\right] \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \\ r_0 \\ \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{pmatrix}$$

$$\underline{u}^i = \underline{\underline{R}}^i \underline{d}^i \quad . \qquad (5.2)$$

The 6 x 6 matrix $\underline{\underline{R}}^i$ is invertible, unless the block has zero area. Defining this inverse to be the matrix $\underline{\underline{Q}}^i$, it follows that:

$$\underline{d}^i = \underline{\underline{Q}}^i \underline{u}^i \quad \text{and} \quad \underline{d}^{iT} = \underline{u}^{iT} \underline{\underline{Q}}^{iT} \qquad (5.3)$$

5.3 Solution Equations for a Single Sub-Block

In this section, the terms that make up the stiffness matrix and load vector for a single block in the DDA method are considered, without interaction with any other block. In general, the total potential energy for a single block, block i, is of the form

$$\Pi = \frac{1}{2}\underline{d}^{i^T}\underline{\underline{K}}^{ii}\underline{d}^{i} - \underline{d}^{i^T}\underline{f}^{i} \quad . \tag{5.4}$$

Minimising this function in terms of the individual block deformation variables and then solving for these unknowns yields

$$\frac{\partial \Pi}{\partial \underline{d}^{i}} = \underline{\underline{K}}^{ii}\underline{d}^{i} - \underline{f}^{i} = \underline{0}$$

$$\underline{d}^{i} = \underline{\underline{K}}^{ii^{-1}}\underline{f}^{i} \quad . \tag{5.5}$$

The total potential energy terms for elastic strain, initial stress, point and volume loads, applied displacements and inertia of a block are summarised in Table 5.1.

By substituting Equation 5.3 into Equation 5.4, a determination of the total potential energy in terms of the nodal degrees of freedom is arrived at:

$$\Pi = \frac{1}{2}\underline{u}^{i^T}\underline{\underline{Q}}^{i^T}\underline{\underline{K}}^{ii}\underline{\underline{Q}}^{i}\underline{u}^{i} - \underline{u}^{i^T}\underline{\underline{Q}}^{i^T}\underline{f}^{i} \quad . \tag{5.6}$$

Minimising with respect to $\underline{u}^{i}$,

$$\frac{\partial \Pi}{\partial \underline{u}^{i}} = \underline{\underline{Q}}^{i^T}\underline{\underline{K}}^{ii}\underline{\underline{Q}}^{i}\underline{u}^{i} - \underline{\underline{Q}}^{i^T}\underline{f}^{i} = \underline{0}$$

$$\underline{u}^{i} = \left[\underline{\underline{Q}}^{i^T}\underline{\underline{K}}^{ii}\underline{\underline{Q}}^{i}\right]^{-1}\left(\underline{\underline{Q}}^{i^T}\underline{f}^{i}\right)$$

$$\underline{u}^{i} = \underline{\underline{K}}'^{ii^{-1}}\underline{f}'^{i} \tag{5.7}$$

where

$$\underline{\underline{K}}'^{ii} = \underline{\underline{Q}}^{i^T}\underline{\underline{K}}^{ii}\underline{\underline{Q}}^{i}$$

$$\underline{f}'^{i} = \underline{\underline{Q}}^{i^T}\underline{f}^{i} \quad . \tag{5.8}$$

Minimisation of the potential energy in terms of the nodal displacements yields the expressions for the stiffness and force arrays given in Table 5.2.

Component	Total Potential Energy (Π)	Minimisation	Global Equations
Elastic Strain	$\Pi_e = \frac{S^i}{2} \underline{d}^{iT} \underline{\underline{E}}\ \underline{d}^i$	$K_{rs}^{ii} = \frac{\partial^2 \Pi_e}{\partial d_r^i . \partial d_s^i}$	$S^i \underline{\underline{E}} \Rightarrow \underline{\underline{K}}^{ii}$
Initial Stress	$\Pi_\sigma = S^i \underline{d}^{iT} \underline{\sigma}_0^i$	$f_r^i = -\frac{\partial \Pi_\sigma}{\partial d_r^i}$	$-S^i(\underline{\sigma}_0) \Rightarrow \underline{f}^i$
Point Loading	$\Pi_p = -\underline{d}^{iT} \underline{\underline{T}}^{iT} \begin{pmatrix} F_x \\ F_y \end{pmatrix}$	$f_r^i = -\frac{\partial \Pi_p}{\partial d_r^i}$	$\underline{\underline{T}}^{iT} \begin{pmatrix} F_x \\ F_y \end{pmatrix} \Rightarrow \underline{f}^i$
Volume Loading	$\Pi_v = -\underline{d}^{iT} \begin{pmatrix} f_x . S^i \\ f_y . S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$f_r^i = -\frac{\partial \Pi_v}{\partial d_r^i}$	$\begin{pmatrix} f_x S^i \\ f_y S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \underline{f}^i$
Assigned Displacements	$\Pi_m = \frac{p_d}{2} \underline{d}^{iT} \underline{\underline{T}}^{iT} \underline{\underline{T}}^i \underline{d}^i$ $- p_d \underline{d}^{iT} \underline{\underline{T}}^{iT} \begin{pmatrix} u_m \\ v_m \end{pmatrix} + \frac{p_d}{2} \begin{pmatrix} u_m & v_m \end{pmatrix} \begin{pmatrix} u_m \\ v_m \end{pmatrix}$	$f_r^i = \frac{\partial}{\partial d_r^i} p_d d_t^i T_{ut}^i \bar{u}_{mu}$ $K_{rs}^{ii} = \frac{\partial^2}{\partial d_r^i \partial d_s^i} \frac{p_d}{2} d_t^i T_{ut}^i T_{uv}^i d_v^i$	$p_d \underline{\underline{T}}^{iT} \begin{pmatrix} u_m \\ v_m \end{pmatrix} \Rightarrow \underline{f}^i$ $p_d \underline{\underline{T}}^{iT} \underline{\underline{T}}^i \Rightarrow \underline{\underline{K}}^{ii}$
Forces of Inertia	$\Pi_i = \underline{d}^{iT} \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^i . dx.dy \right) \frac{2M}{\Delta t^2} . \underline{d}^i$ $- \underline{d}^{iT} \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^i . dx.dy \right) \frac{2M}{\Delta t} \underline{\mathbf{v}}^i(0)$	$f_r^i = -\frac{\partial}{\partial d_r^i} d_t^i \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^i . dx.dy \right)_{tu} \frac{2M}{\Delta t} \underline{\mathbf{v}}^i(0)_u$ $K_{rs}^{ii} = \frac{\partial^2}{\partial d_r^i \partial d_s^i} d_t^i \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^i . dx.dy \right)_{tu} \frac{2M}{\Delta t^2} d_u^i$	$\frac{2M}{\Delta t} \iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^i . dx.dy\ \underline{\mathbf{v}}_0^i \Rightarrow \underline{f}^i$ $\frac{2M}{\Delta t^2} \iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^i . dx.dy \Rightarrow \underline{\underline{K}}^{ii}$

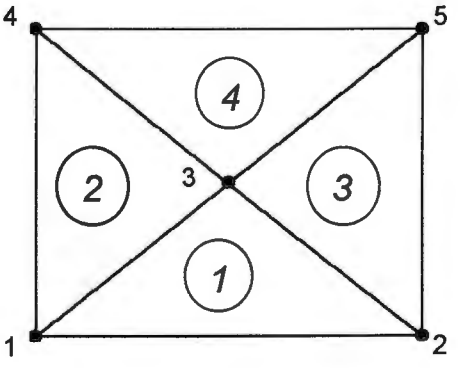
Table 5.1: Summary of TPE minimisation for a single block in the DDA method.

Component	Total Potential Energy (Π)	Minimisation	Global Equations
Elastic Strain	$\Pi_e = \frac{S^i}{2} \underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{\underline{E}}\ \underline{\underline{Q}}^{i} \underline{u}^{i}$	$K_{rs}^{ii} = \frac{\partial^2 \Pi_e}{\partial u_r^i . \partial u_s^i}$	$S^i \underline{\underline{Q}}^{iT} \underline{\underline{E}}\ \underline{\underline{Q}}^{i} \Rightarrow \underline{\underline{K}}^{ii}$
Initial Stress	$\Pi_\sigma = S^i \underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{\sigma}_0^i$	$f_r^i = -\frac{\partial \Pi_\sigma}{\partial u_r^i}$	$-S^i \underline{\underline{Q}}^{iT} \left(\underline{\sigma}_0^i\right) \Rightarrow \underline{f}^i$
Point Loading	$\Pi_p = -\underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{\underline{T}}^{iT} \begin{pmatrix} F_x \\ F_y \end{pmatrix}$	$f_r^i = -\frac{\partial \Pi_p}{\partial u_r^i}$	$\underline{\underline{Q}}^{iT} \underline{\underline{T}}^{iT} \begin{pmatrix} F_x \\ F_y \end{pmatrix} \Rightarrow \underline{f}^i$
Volume Loading	$\Pi_v = -\underline{u}^{iT} \underline{\underline{Q}}^{iT} \begin{pmatrix} f_x S^i \\ f_y S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$	$f_r^i = -\frac{\partial \Pi_v(0)}{\partial u_r^i}$	$\underline{\underline{Q}}^{iT} \begin{pmatrix} f_x S^i \\ f_y S^i \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \Rightarrow \underline{f}^i$
Assigned Displacements	$\Pi_m = \frac{p_d}{2} \underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} \underline{\underline{Q}}^{i} \underline{u}^{i} -$ $p_d \underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{\underline{T}}^{iT} \begin{pmatrix} u_m \\ v_m \end{pmatrix} + \frac{p_d}{2} \begin{pmatrix} u_m & v_m \end{pmatrix} \begin{pmatrix} u_m \\ v_m \end{pmatrix}$	$f_r^i = \frac{\partial}{\partial u_r^i} p u_{tt}^i Q_{ut}^i T_{vu}^i \bar{u}_{mv}$ $K_{rs}^{ii} = \frac{\partial^2}{\partial u_r^i . \partial u_s^i} \frac{p_d}{2} u_t^i Q_{ut}^i T_{vu}^i T_{vw}^i Q_{wz}^i u_z^i$	$p\ \underline{\underline{Q}}^{iT} \underline{\underline{T}}^{iT} \begin{pmatrix} u_m \\ v_m \end{pmatrix} \Rightarrow \underline{f}^i$ $p\ \underline{\underline{Q}}^{iT} \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} \underline{\underline{Q}}^{i} \Rightarrow \underline{\underline{K}}^{ii}$
Forces of Inertia	$\Pi_i = \underline{u}^{iT} \underline{\underline{Q}}^{iT} \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} . dx.dy \right) \frac{2M}{\Delta t^2} \underline{\underline{Q}}^{i} \underline{u}^{i}$ $-\underline{u}^{iT} \underline{\underline{Q}}^{iT} \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} . dx.dy \right) \frac{2M}{\Delta t} \underline{\mathbf{v}}^i(0)$	$f_r^i = -\frac{\partial}{\partial u_r^i} u_t^i Q_{ut}^i \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} . dx.dy \right)_{uv} \frac{2M}{\Delta t} \underline{\mathbf{v}}^i(0)$ $K_{rs}^{ii} = \frac{\partial^2}{\partial u_r^i \partial u_s^i} u_t^i Q_{ut}^i \left(\iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} . dx.dy \right)_{uv} \frac{2M}{\Delta t^2} Q_{vw}^i u_w^i$	$\frac{2M}{\Delta t} \underline{\underline{Q}}^{iT} \iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} . dx.dy \underline{\mathbf{v}}^i(0) \Rightarrow \underline{f}^i$ $\frac{2M}{\Delta t^2} \underline{\underline{Q}}^{iT} \iint \underline{\underline{T}}^{iT} \underline{\underline{T}}^{i} . dx.dy\ \underline{\underline{Q}}^{iT} \Rightarrow \underline{\underline{K}}^{i}$

Table 5.2: Summary of TPE minimisation for a single block in terms of nodal displacements.

5.4 Solution Equations for a Sub-meshed Block

If all of the sub-block sub-matrices of a block are transformed by this procedure, then the global stiffness matrix is now written in terms of the nodal displacements, as it would be in a finite element analysis. However, neighbouring elements share the same nodes, and the terms in the stiffness matrix can be combined for these shared nodes. In the simple example of Figure 5.2, the block stiffness matrices of sub-blocks 1 and 2 are inserted into the global stiffness matrix in the positions shown:



$$\begin{bmatrix}
K^{11}_{11}+K^{22}_{11} & K^{11}_{12}+K^{22}_{12} & K^{11}_{13} & K^{11}_{14} & K^{11}_{15}+K^{22}_{13} & K^{11}_{16}+K^{22}_{14} & K^{22}_{15} & K^{22}_{16} & - & - \\
K^{11}_{21}+K^{22}_{21} & K^{11}_{22}+K^{22}_{22} & K^{11}_{23} & K^{11}_{24} & K^{11}_{25}+K^{22}_{23} & K^{11}_{26}+K^{22}_{24} & K^{22}_{25} & K^{22}_{26} & - & - \\
K^{11}_{31} & K^{11}_{32} & K^{11}_{33} & K^{11}_{34} & K^{11}_{35} & K^{11}_{36} & - & - & - & - \\
K^{11}_{41} & K^{11}_{42} & K^{11}_{43} & K^{11}_{44} & K^{11}_{45} & K^{11}_{46} & - & - & - & - \\
K^{11}_{51}+K^{22}_{31} & K^{11}_{52}+K^{22}_{32} & K^{11}_{53} & K^{11}_{54} & K^{11}_{55}+K^{22}_{33} & K^{11}_{56}+K^{22}_{34} & K^{22}_{35} & K^{22}_{36} & - & - \\
K^{11}_{61}+K^{22}_{41} & K^{11}_{62}+K^{22}_{42} & K^{11}_{63} & K^{11}_{64} & K^{11}_{65}+K^{22}_{43} & K^{11}_{66}+K^{22}_{44} & K^{22}_{45} & K^{22}_{46} & - & - \\
K^{22}_{51} & K^{22}_{52} & - & - & K^{22}_{53} & K^{22}_{54} & K^{22}_{55} & K^{22}_{56} & - & - \\
K^{22}_{61} & K^{22}_{62} & - & - & K^{22}_{63} & K^{22}_{64} & K^{22}_{65} & K^{22}_{66} & - & - \\
- & - & - & - & - & - & - & - & - & - \\
- & - & - & - & - & - & - & - & - & -
\end{bmatrix}
\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ u_3 \\ v_3 \\ u_4 \\ v_4 \\ u_5 \\ v_5 \end{pmatrix}
=
\begin{pmatrix} f^{11}_1+f^{12}_1 \\ f^{11}_2+f^{12}_2 \\ f^{11}_3 \\ f^{11}_4 \\ f^{11}_5+f^{12}_3 \\ f^{11}_6+f^{12}_4 \\ f^{12}_5 \\ f^{12}_6 \\ - \\ - \end{pmatrix}$$

(Only terms for elements 1 and 2 have been added to the global stiffness matrix)

Figure 5.2: Solution equations for a simple mesh.

When the terms from elements 3 and 4 have been added, the above matrix may be inverted, and the system solved. The block displacement unknowns may then be obtained by applying Equation 5.3.

A further note should be added. In the DDA method, prescribed displacements are enforced by means of a penalty method. This procedure may still be used in the method described above. Alternatively, if the prescribed displacements occur at one of the nodes of the mesh, they may be added after the assembly of the global stiffness

matrix. The method for doing this is identical to that used in the Finite Element method. In the simple global stiffness matrix below, if node 1 has a prescribed displacement of v_m in the *y*-direction, then the solution equations are altered before solution to the form shown:

$$\begin{bmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \\ K_{31} & K_{32} & K_{33} & K_{34} \\ K_{41} & K_{42} & K_{43} & K_{44} \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix} \Rightarrow \begin{bmatrix} K_{11} & 0 & K_{13} & K_{14} \\ 0 & 1 & 0 & 0 \\ K_{31} & 0 & K_{33} & K_{34} \\ K_{41} & 0 & K_{43} & K_{44} \end{bmatrix} \begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} f_1 - K_{12} v_m \\ v_m \\ f_3 - K_{32} v_m \\ f_4 - K_{42} v_m \end{pmatrix}$$

This method eliminates the small errors obtained when the penalty method is used. However, it imposes the restriction that, in sub-meshed blocks, displacements may only be prescribed at nodal points.

Similarly, point loads can be applied after assembly of the global matrices, simply by adding them to the appropriate degree of freedom in the force array. It will be found, however, that if point loads are applied using the standard DDA method, an identical result is obtained after transformation.

5.5 Block Interaction

The foregoing theory has described how a single, sub-divided block may be solved as a Finite Element mesh, rather than a DDA problem. However, interactions between blocks must also be considered. It is assumed that in the solution process only blocks that are sub-meshed are transformed to Finite Element form solutions. There is no need to transform a single-element block, and if this is done one loses the advantage that a DDA block may be of any polygonal shape, whereas the sub-blocks considered here must be triangular.

There are two types of interactions between blocks in DDA, namely bolting connections and contacts. For each, there are now three possibilities: a DDA-type block can interact with another DDA block or a FEM element, or two FEM elements may interact. In the first case, the standard DDA method prevails. But where transformed blocks are involved, the minimisation of the total potential energy must be reconsidered.

In Chapter 3, it was shown (Equation 3.36) that the potential energy of a bolted connection is

$$\Pi_b = \frac{p_b}{2l}\left(\underline{d}^{iT}\,\underline{e}\;\underline{e}^T\underline{d}^i - \underline{d}^{iT}\,\underline{e}\;\underline{g}^T\underline{d}^j - \underline{d}^{jT}\,\underline{g}\;\underline{e}^T\underline{d}^i + \underline{d}^{jT}\,\underline{g}\;\underline{g}^T\underline{d}^j\right) \quad .$$

For a normal spring (Equation 3.43),

$$\Pi_p = \frac{p_c}{2}\left(\frac{S^2}{l^2} + \underline{d}^{i^T}\underline{e}\,\underline{e}^T\underline{d}^i + \underline{d}^{j^T}\underline{g}\,\underline{g}^T\underline{d}^j + \underline{d}^{i^T}\underline{e}\,\underline{g}^T\underline{d}^j + \underline{d}^{j^T}\underline{g}\,\underline{e}^T\underline{d}^i + \frac{2S}{l}\underline{e}^T\underline{d}^i + \frac{2S}{l}\underline{g}^T\underline{d}^j\right) \quad .$$

For a shear spring (Equation 3.52),

$$\Pi_k = \frac{p_c}{2}\left(\frac{S^2}{l^2} + \underline{d}^{i^T}\underline{e}\,\underline{e}^T\underline{d}^i + \underline{d}^{j^T}\underline{g}\,\underline{g}^T\underline{d}^j + \underline{d}^{i^T}\underline{e}\,\underline{g}^T\underline{d}^j + \underline{d}^{j^T}\underline{g}\,\underline{e}^T\underline{d}^i + \frac{2S}{l}\underline{e}^T\underline{d}^i + \frac{2S}{l}\underline{g}^T\underline{d}^j\right) \quad .$$

Finally, for the friction force (Equation 3.60),

$$\Pi_f = \tilde{F}_s\frac{S}{l} + \tilde{F}_s\underline{e}^T\underline{d}^i + \tilde{F}_s\underline{g}^T\underline{d}^j \quad .$$

The terms for $\underline{e}$, g and S are different for the different equations. Only the transformation of a normal spring will be considered in this report. The derivations for the other three equations are similar.

If both blocks are DDA blocks, then the energy equation is unaffected by the meshing elsewhere in the system.

If both blocks are FEM elements then it is important to note that they are elements of different meshes. There is no bolting or contact interaction within a mesh because a mesh represents a single block. Equation 5.3 is substituted into the total potential energy equation and minimised with respect to nodal displacements.

$$\Pi_p = \frac{p_c}{2}\left(\frac{S^2}{l^2} + \underline{u}^{i^T}\underline{\underline{Q}}^{i^T}\underline{e}\,\underline{e}^T\underline{\underline{Q}}^i\underline{u}^i + \underline{u}^{j^T}\underline{\underline{Q}}^{j^T}\underline{g}\,\underline{g}^T\underline{\underline{Q}}^j\underline{u}^j + \underline{u}^{i^T}\underline{\underline{Q}}^{i^T}\underline{e}\,\underline{g}^T\underline{d}^j + \underline{u}^{j^T}\underline{\underline{Q}}^{j^T}\underline{g}\,\underline{e}^T\underline{\underline{Q}}^i\underline{u}^i + \frac{2S}{l}\underline{e}^T\underline{\underline{Q}}^i\underline{u}^i + \frac{2S}{l}\underline{g}^T\underline{\underline{Q}}^j\underline{u}^j\right) \qquad (5.9)$$

$$K'^{ii}_{rs} = \frac{\partial^2\Pi_p}{\partial u^i_r.\partial u^i_s} = \frac{p_c}{2}\frac{\partial^2}{\partial u^i_r.\partial u^i_s}\left(\underline{u}^{i^T}\underline{\underline{Q}}^{i^T}\underline{e}\,\underline{e}^T\underline{\underline{Q}}^i\underline{u}^i\right) = p_c Q^i_{mr}e_m e_n Q^i_{ns}$$

$$p_c \underline{\underline{Q}}^{iT} \underline{e}\ \underline{e}^T \underline{\underline{Q}}^i \Rightarrow \underline{\underline{K}}'^{ii} \tag{5.10}$$

$$K'^{ij}_{rs} = \frac{\partial^2 \Pi_p}{\partial u^i_r . \partial u^j_s} = \frac{p_c}{2} \frac{\partial^2}{\partial u^i_r . \partial u^j_s} \left(\underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{e}\ \underline{g}^T \underline{\underline{Q}}^j \underline{u}^j + \underline{u}^{jT} \underline{\underline{Q}}^{jT} \underline{g}\ \underline{e}^T \underline{\underline{Q}}^i \underline{u}^i \right)$$
$$= p_c Q^i_{mr} e_m g_n Q^j_{ns}$$

$$p_c \underline{\underline{Q}}^{iT} \underline{e}\ \underline{g}^T \underline{\underline{Q}}^j \Rightarrow \underline{\underline{K}}'^{ij} \tag{5.11}$$

$$K'^{ji}_{rs} = \frac{\partial^2 \Pi_p}{\partial u^j_r . \partial u^i_s} = \frac{p_c}{2} \frac{\partial^2}{\partial u^j_r . \partial u^i_s} \left(\underline{u}^{jT} \underline{\underline{Q}}^{jT} \underline{g}\ \underline{e}^T \underline{\underline{Q}}^i \underline{u}^i + \underline{u}^{iT} \underline{\underline{Q}}^{iT} \underline{e}\ \underline{g}^T \underline{\underline{Q}}^j \underline{u}^j \right)$$
$$= p_c Q^j_{mr} g_m e_n Q^i_{ns}$$

$$p_c \underline{\underline{Q}}^{jT} \underline{g}\ \underline{e}^T \underline{\underline{Q}}^i \Rightarrow \underline{\underline{K}}'^{ji} \tag{5.12}$$

$$K'^{jj}_{rs} = \frac{\partial^2 \Pi_p}{\partial u^j_r . \partial u^j_s} = \frac{p_c}{2} \frac{\partial^2}{\partial u^j_r . \partial u^j_s} \left(\underline{u}^{jT} \underline{\underline{Q}}^{jT} \underline{g}\ \underline{g}^T \underline{\underline{Q}}^j \underline{u}^j \right) = p_c Q^j_{mr} g_m g_n Q^j_{ns}$$

$$p_c \underline{\underline{Q}}^{jT} \underline{g}\ \underline{g}^T \underline{\underline{Q}}^j \Rightarrow \underline{\underline{K}}'^{jj} \tag{5.13}$$

$$f^i_r = -\frac{\partial}{\partial u^i_r} \left(\frac{p_c S}{l} \underline{e}^T \underline{\underline{Q}}^i \underline{u}^i \right) = -\frac{p_c S}{l} e_m Q^i_{mr}$$

$$-\frac{p_c S}{l} \underline{\underline{Q}}^{iT} \underline{e} \Rightarrow \underline{f}^i \tag{5.14}$$

$$f^j_r = -\frac{\partial}{\partial u^j_r} \left(\frac{p_c S}{l} \underline{g}^T \underline{\underline{Q}}^j \underline{u}^j \right) = -\frac{p_c S}{l} g_m Q^j_{mr}$$

$$-\frac{p_c S}{l} \underline{\underline{Q}}^{jT} \underline{g} \Rightarrow \underline{f}^j \tag{5.15}$$

The submatrices of Equations 5.10 to 5.15 are substituted into their respective positions in the global solution equations.

If a DDA block, block i, interacts with a FEM element, element j, then the formulation is similar. Substituting Equation 5.3 into the total potential energy equation to convert to the nodal degrees of freedom of element j, but leaving block i with its original block displacement variables,

$$\Pi_p = \frac{p_c}{2}\left(\frac{S^2}{l^2} + \underline{d}^{iT}\,\underline{e}\,\underline{e}^T\,\underline{d}^i + \underline{u}^{jT}\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{g}^T\,\underline{\underline{Q}}^j\,\underline{u}^j + \underline{d}^{iT}\,\underline{e}\,\underline{g}^T\,\underline{d}^j + \underline{u}^{jT}\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{e}^T\,\underline{d}^i + \frac{2S}{l}\underline{e}^T\,\underline{d}^i + \frac{2S}{l}\underline{g}^T\,\underline{\underline{Q}}^j\,\underline{u}^j\right) \quad . \tag{5.16}$$

This function is minimised as before:

$$K'^{ii}_{rs} = \frac{\partial^2 \Pi_p}{\partial d^i_r . \partial d^i_s} = \frac{p_c}{2}\frac{\partial^2}{\partial d^i_r . \partial d^i_s}\left(\underline{d}^{iT}\,\underline{e}\,\underline{e}^T\,\underline{d}^i\right) = p_c e_r e_s$$

$$p_c\,\underline{e}\,\underline{e}^T \Rightarrow \underline{\underline{K}}'^{ii} \tag{5.17}$$

$$K'^{ij}_{rs} = \frac{\partial^2 \Pi_p}{\partial d^i_r . \partial u^j_s} = \frac{p_c}{2}\frac{\partial^2}{\partial d^i_r . \partial u^j_s}\left(\underline{d}^{iT}\,\underline{e}\,\underline{g}^T\,\underline{\underline{Q}}^j\,\underline{u}^j + \underline{u}^{jT}\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{e}^T\,\underline{d}^i\right) = p_c e_r g_n Q^j_{ns}$$

$$p_c\,\underline{e}\,\underline{g}^T\,\underline{\underline{Q}}^j \Rightarrow \underline{\underline{K}}'^{ij} \tag{5.18}$$

$$K'^{ji}_{rs} = \frac{\partial^2 \Pi_p}{\partial u^j_r . \partial d^i_s} = \frac{p_c}{2}\frac{\partial^2}{\partial u^j_r . \partial d^i_s}\left(\underline{u}^{jT}\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{e}^T\,\underline{d}^i + \underline{d}^{iT}\,\underline{e}\,\underline{g}^T\,\underline{\underline{Q}}^j\,\underline{u}^j\right) = p_c Q^j_{mr} g_m e_s$$

$$p_c\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{e}^T \Rightarrow \underline{\underline{K}}'^{ji} \tag{5.19}$$

$$K'^{jj}_{rs} = \frac{\partial^2 \Pi_p}{\partial u^j_r . \partial u^j_s} = \frac{p_c}{2}\frac{\partial^2}{\partial u^j_r . \partial u^j_s}\left(\underline{u}^{jT}\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{g}^T\,\underline{\underline{Q}}^j\,\underline{u}^j\right) = p_c Q^j_{mr} g_m g_n Q^j_{ns}$$

$$p_c\,\underline{\underline{Q}}^{jT}\,\underline{g}\,\underline{g}^T\,\underline{\underline{Q}}^j \Rightarrow \underline{\underline{K}}'^{jj} \tag{5.20}$$

$$f_r^i = -\frac{\partial \Pi_p}{\partial d_r^i} = -\frac{p_c S}{l}\frac{\partial}{\partial d_r^i}\left(\underline{e}^T \underline{d}^i\right) = -\frac{p_c S}{l} e_r$$

$$-\frac{p_c S}{l}\underline{e} \Rightarrow \underline{f}^i \qquad (5.21)$$

$$f_r^j = -\frac{\partial \Pi_p}{\partial u_r^j} = -\frac{p_c S}{l}\frac{\partial}{\partial u_r^j}\left(\underline{g}^T \underline{\underline{Q}}^j \underline{u}^j\right) = -\frac{p_c S}{l} g_m Q_{mr}^j$$

$$-\frac{p_c S}{l}\underline{\underline{Q}}^{jT}\underline{g} \Rightarrow \underline{f}^j \qquad (5.22)$$

The sub-matrices of Equations 5.17 to 5.22 are added to their respective positions in the global solution equations.

5.6 Formation of the Global Solution Equations

From the above derivations, a simple transformation procedure may be formulated. In a problem with a total of n blocks and sub-blocks, there will be n^2 block stiffness matrices, $\underline{\underline{K}}^{ij}$ where i=1..n and j=1..n. These are formulated by the standard DDA method. Then for each sub-block m, all block stiffness matrices $\underline{\underline{K}}^{mj}$ are pre-multiplied by matrix $\underline{\underline{Q}}^{m\,T}$. Similarly, all block stiffness matrices $\underline{\underline{K}}^{im}$ are post-multiplied by matrix $\underline{\underline{Q}}^m$. The diagonal term $\underline{\underline{K}}^{mm}$ will be pre- and post-multiplied in this process. Finally, the block force vector $\underline{f}^m$ is pre-multiplied by matrix $\underline{\underline{Q}}^{m\,T}$.

$$\underline{\underline{K}}'^{mj} = \underline{\underline{Q}}^{mT}\,\underline{\underline{K}}^{mj}$$

$$\underline{\underline{K}}'^{im} = \underline{\underline{K}}^{im}\,\underline{\underline{Q}}^{m}$$

$$\underline{f}'^{m} = \underline{\underline{Q}}^{mT}\,\underline{f}^{m}$$

where $i = 1,..,n$ $\quad$ $j = 1,..,n$.

The procedure is repeated for each sub-block element.

When this transformation process has been completed, the transformed terms in the stiffness matrix and force vector must be distributed by row and column to the correct nodal degrees of freedom in the global displacement vector.

The overall result is a matrix of the form shown in Figure 5.3. Sub-blocks 4, 5 and 6 are meshed into a single block, while blocks 1, 2, 3 and 7 are standard DDA blocks. The resulting matrix is symmetric and invertible.

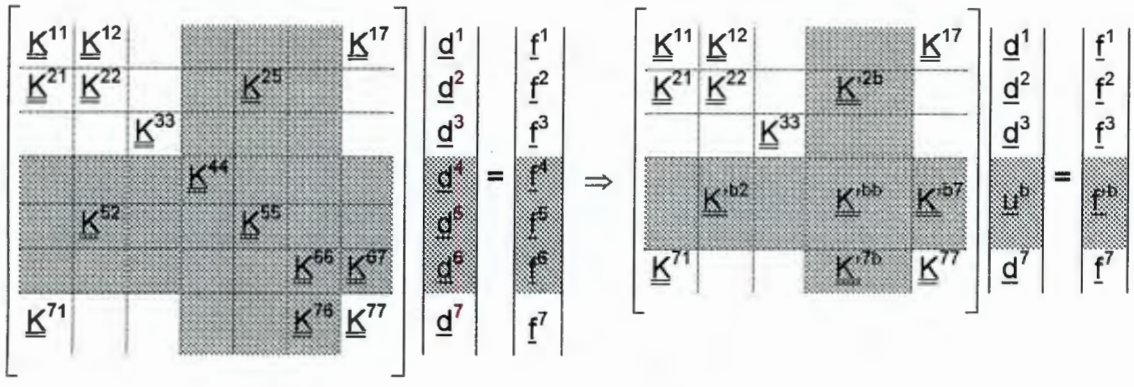


Figure 5.3: Overall format for conversion of global equations.

5.7 Testing of the Meshing Formulation

Due to time constraints, it was not possible to alter the original DDA code to allow sub-meshing. Instead, the formulation described above was tested using Mathcad Plus 6.0 [1995].

A simplified DDA code was written on Mathcad Plus 6.0. This code concentrates on the formulation and solution of the equilibrium equations for block systems, and only analyses a single time increment.

A simple dynamic problem was then set up, with loads acting on two triangular blocks which are in contact with each other. The problem was analysed using both the Mathcad code, and DDA Version 96 [1996]. The results obtained were found to be identical (see Appendix A).

Triangular blocks can be treated either as standard DDA blocks, or as single sub-block elements. Therefore, in the Mathcad code, the solution equations from the standard DDA method were transformed so as to solve for the deformation parameters of the first block, and the nodal displacements of the second block. The transformed solution equations were solved and the deformation parameters of the second block recovered. The results were identical to those of the previous analyses.

Finally, the original solution equations were transformed so that they were in terms of the nodal displacements of both blocks. Again, the solution obtained was identical to those of the other analyses.

Details of all of these analyses and the Mathcad Plus 6.0 codes are contained in Appendix A.

Similar analyses were conducted on two triangular blocks connected by a rock bolt. All of these tests verify that the sub-meshing transformation procedure is mathematically correct, and that the solution equations for a block system may be converted to solve for different combinations of DDA blocks and sub-block elements, without affecting the results obtained.

A separate code was then developed on Mathcad Plus 6.0, loosely based upon the first code. This code was designed to analyse a single sub-meshed block, without interaction with other blocks. Because only a single time increment can be modelled, tests were practically limited to static analyses, and so inertia effects were not considered. The code was used to analyse the problems described in Section 5.8.

This second code is reproduced in Appendix B.

5.8 Performance of 1st Order Element Meshes

All of the tests in this section were performed using the code described in Section 5.7, and printed in Appendix B. For all tests, the Young's modulus and Poisson's ratio of the material were set at 40 GPa and 0.3, respectively.

5.8.1 Convergence criteria and the patch test

A Finite Element simulation is in general an approximate solution to a real problem. It is required that as a mesh is refined by decreasing the size of the elements, the Finite Element solution will converge on a correct solution. There are two criteria which guarantee that this will be the case. These are the continuity condition and the completeness condition.

The continuity condition requires that the displacement solution should be continuous at element boundaries. This means, in effect, that neighbouring elements may not separate or overlap along their common edges.

The 1st order elements described in this chapter have linear displacement functions. As a result of this, a straight line between two points will remain straight after displacement. If the elements are meshed corner-to-corner, then neighbouring

elements will share two common nodes, and their common edge will be the straight line between these two nodes. With correct meshing, therefore, the elements conform to the continuity condition.

The completeness condition states that the displacement function of an element and its first derivatives should be able to assume any constant value.

If the displacement function is constant, then this corresponds to a rigid body motion of the element. It is necessary that the element remains free of strains when such a displacement is applied.

Rigid body motions were among the original deformation degrees of freedom of the DDA element. It would be expected, therefore, that strains would be zero when a constant displacement function is applied. Tests show that this is the case.

The strains in an element are functions of the first derivatives of the displacement function. If a uniform, linear displacement function is imposed upon an element, it is required that the strains are constant throughout the element. The justification for this is that, in the limit of mesh refinement, the elements become infinitesimally small. Therefore, the strain throughout a particular element is approximately constant. If an element is able to assume this constant strain, then, in the limit, a correct solution will be obtained.

The standard test for this condition is the patch test. A simple mesh, or patch, of elements is formulated, the elements being irregularly shaped. At least one node must lie within the boundary of the patch. A uniform displacement function is then applied to all boundary nodes of the patch. The requirements of the test are that the internal nodes displace in conformity with the general displacement function, and that the derived strains everywhere in the patch are constant.

For more information on convergence criteria and the patch test, the reader is referred to de Arantes and Oliviera [1977].

A patch test was performed on the 1st order element, as depicted in Figure 5.4. The element passed the patch test.

The convergence criteria are therefore satisfied, and the element will converge to a correct solution. However the convergence criteria do not give any indication of the performance of the element in coarse meshes, and further tests are necessary to evaluate this.

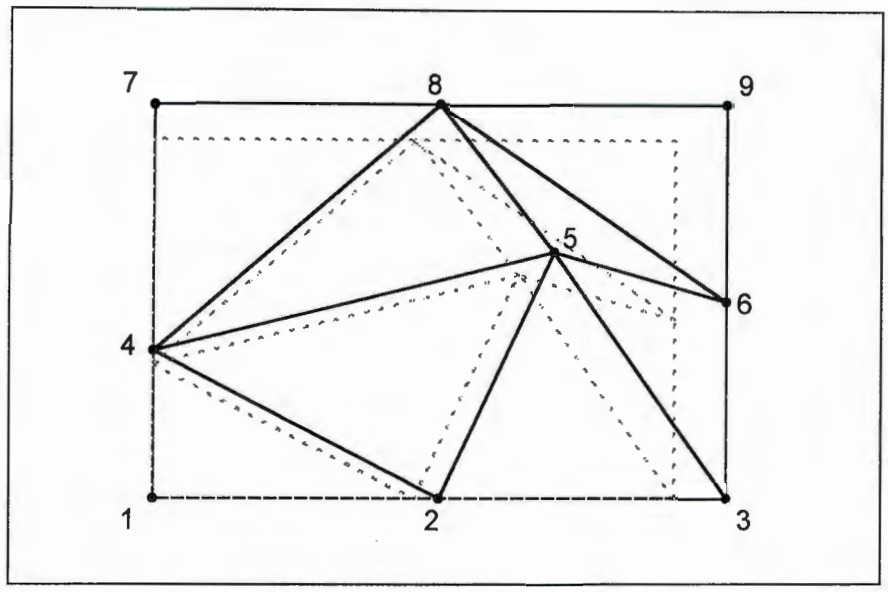


Figure 5.4: The patch test.

5.8.2 Varying stress field

It is an assumption of the 1st order displacement function that stresses and strains are constant throughout the area of an element. It is questionable, therefore, how well the element will model problems where the stresses and strains vary across the mesh.

A simple test was devised to investigate this. Figure 5.5(a) shows the mesh and boundary conditions used in the test. The only load acting on the mesh is its own self-weight. The body is a cube of unit dimensions, and the density is also set at unity. The problem was analysed and the stresses obtained are plotted in Figure 5.5(b). The stresses in each element are plotted against the depth of the centroid of the element below the top edge. The dotted lines represent the theoretical solution.

The mesh of 1st order elements accurately models the stresses in the body. It can be shown [Barlow, 1976] that in elements of this type, the stresses are determined most accurately at the centroid of the element.

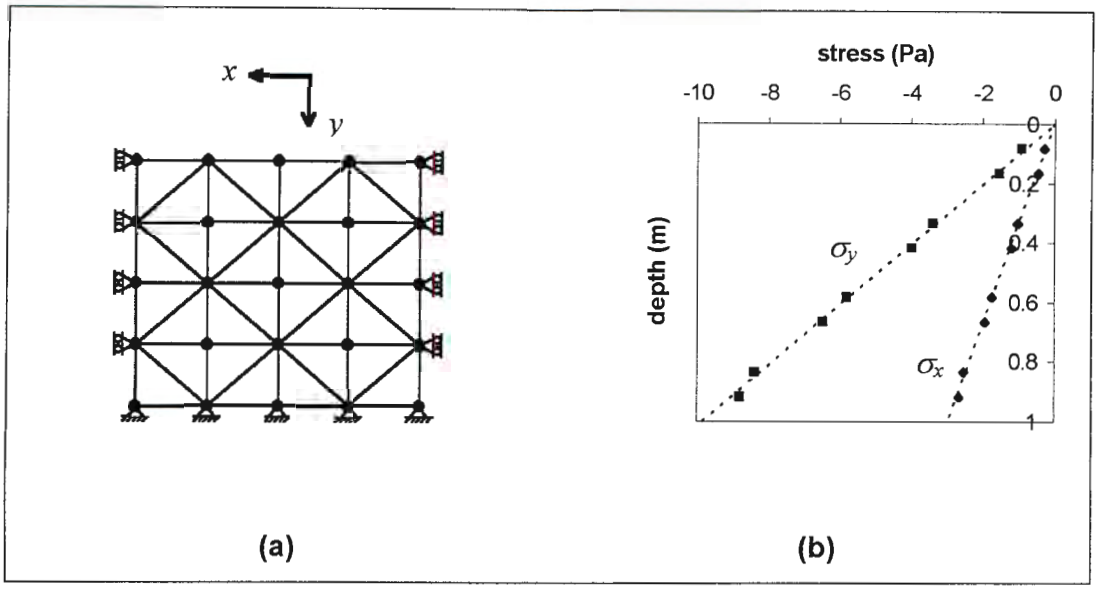


Figure 5.5: Stresses in a mesh loaded by self-weight.

5.8.3 Cook cantilever test

A common test for elements is described by Cook *et al.* [1989]. This consists of a short, tapered cantilever which is subjected to a uniformly distributed load at its free end. The geometry of the problem is shown in Figure 5.6. Also shown is the displaced shape as modelled by a 32-element mesh. The displacements are exaggerated for clarity.

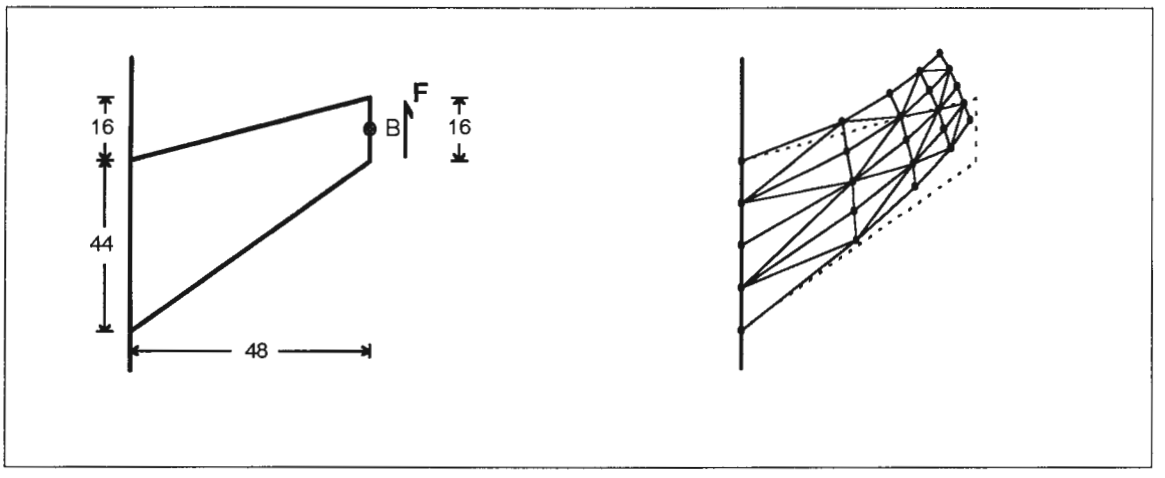


Figure 5.6: Geometry and meshing of the Cook cantilever test.

A variety of meshes of increasing refinement were tested using this problem.

The results of this test series are plotted in Figure 5.7. The vertical displacement of the midpoint of the free face, v_B, is compared for the various meshes. No closed-form solution exists for this problem. The results were therefore compared to a 'best known value', v_{B0}, which is the result obtained from a Finite Element analysis, using 100 8-

noded reduced integration elements. Also plotted in Figure 5.7 are results obtained using Finite Element analysis, with 3-noded 'constant strain triangle' (CST) elements. The Finite Element analyses were performed using the ABAQUS Finite Element program [1996].

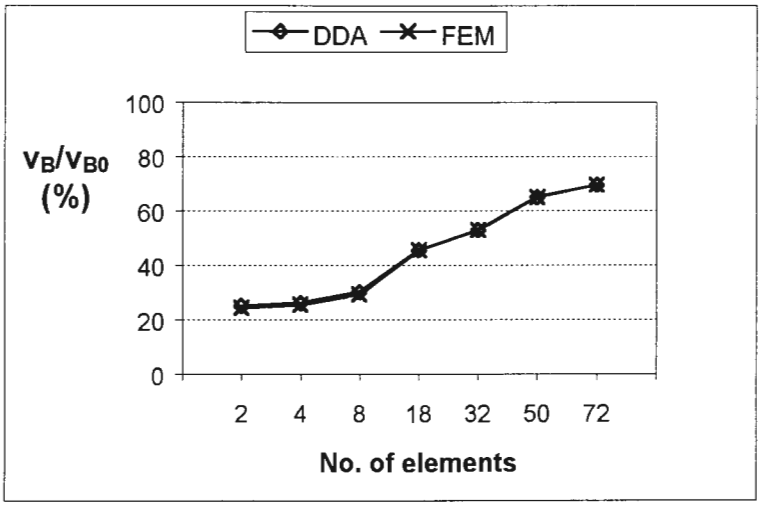


Figure 5.7: Results of the Cook cantilever test for a series of meshes.

It can be seen that the performance of both methods is poor in this test. CST elements are the simplest 2-dimensional elements available in the FE method, and they have long been noted for their poor performance in bending. This is because, as their name suggests, they make the assumption of constant strain across the element. In a member under pure bending, the strain in the longitudinal direction varies linearly across the section, and is zero at the neutral axis. The element cannot model this strain gradient, and as a result it is too stiff in coarse meshes.

Because DDA also makes the assumption of constant strain, the two elements are identical, both being 1st order elements with six degrees of freedom.

5.8.4 Uniaxial compression test

This test models a block under uniaxial compression between two loading plattens. Only one quarter of the plate is modelled, taking advantage of symmetry. The mesh is shown in Figure 5.8. The displaced shape of the block is also shown after the prescribed displacement has been applied. This displacement is exaggerated in the diagram for clarity.

The mesh has 38 nodes, and hence 76 degrees of freedom, and is composed of 54 1st order elements. The original number of block deformation variables for the sub-block elements is therefore 324. This illustrates the considerable reduction in the total number of degrees of freedom of a mesh when its displacement is described in terms of nodal displacements, rather than block deformation parameters. It can be shown

that, if a mesh is uniformly discretised, then in the limit as the number of elements in the mesh approaches infinity, the ratio of nodal displacement degrees of freedom to block deformation degrees of freedom approaches 1 : 6.

A vertical prescribed displacement is applied to the top edge of the mesh of Figure 5.8, equivalent to a vertical strain of 1% in the plate. To simulate friction between the block and the loading plattens, the top edge is restrained horizontally. The Von Mises stresses are derived for each block. Von Mises stress is defined as

$$\sigma_{VM} = \sqrt{\frac{3}{2}\left[(\sigma_x - p)^2 + (\sigma_y - p)^2 + (\sigma_z - p)^2 + \tau_{xy}^{\ 2} + \tau_{xz}^{\ 2} + \tau_{yz}^{\ 2}\right]}$$

where

$$p = \frac{1}{3}(\sigma_x + \sigma_y + \sigma_z)$$

and

$$\sigma_z = \tau_{xz} = \tau_{yz} = 0$$

because plane stress conditions are assumed.

The Von Mises stresses are displayed as a contour plot in Figure 5.9. Stresses at nodes are calculated as the average of the stresses in the blocks bordering on that node. Stresses at other points are then found by linear interpolation. Figure 5.10 shows a contour plot of the same problem, modelled as a Finite Element problem using the ABAQUS Finite Element program [1996], with constant strain triangle first order elements. Contours in the range 380 MPa to 490 MPa are plotted. As with the Cook cantilever test, the results are identical.

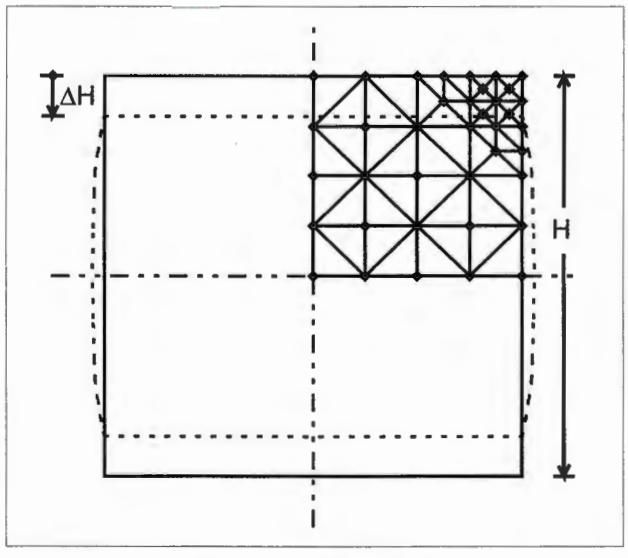


Figure 5.8: Geometry and meshing of the uniaxial compression test.

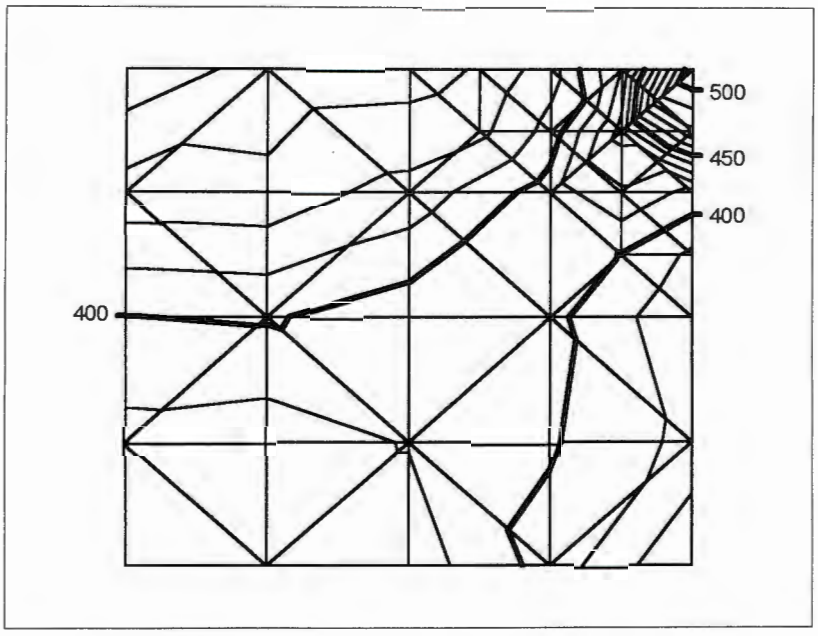


Figure 5.9: DDA solution: Contour plot of Von Mises stresses (MPa) in the uniaxial compression test.

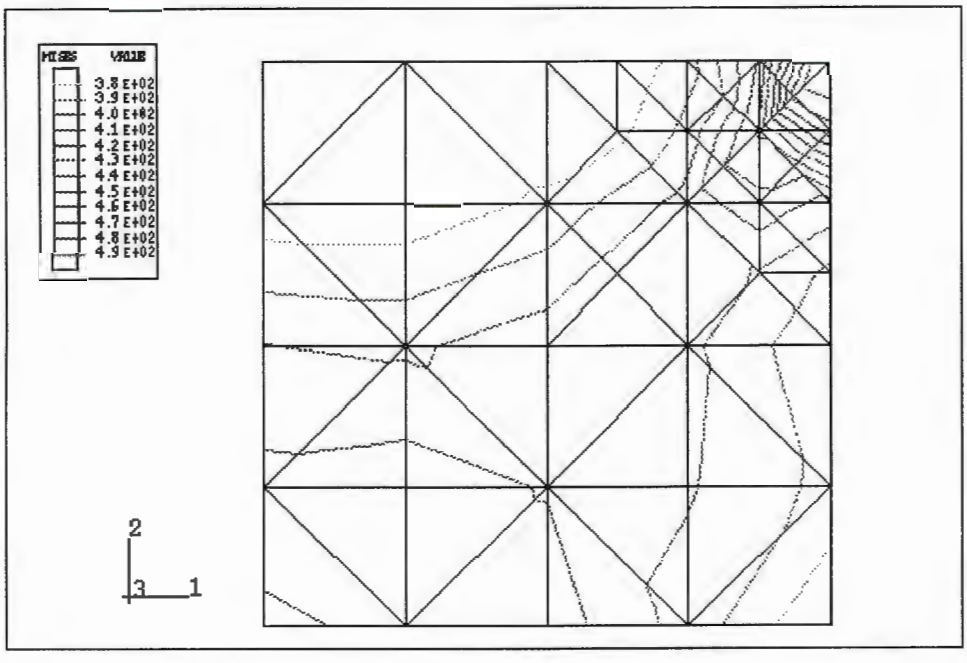


Figure 5.10: FEM solution: Contour plot of Von Mises stresses in the uniaxial compression test. Contours are in the range 380 – 490 MPa.

Chapter 6

MESHING USING 2ND ORDER ELEMENTS

The 1st order linear triangle developed in the previous chapter is the most basic two-dimensional element available for meshing. While this element satisfies the conditions for completeness, it often does not provide accurate results when used in coarse meshes.

In this chapter, meshing using 2nd order elements is discussed. 2nd order blocks have been developed and incorporated into the DDA method previously [Chern *et al.*, 1995]. These blocks may be transformed into 2nd order elements.

The general form of a complete quadratic polynomial displacement function would be:

$$\begin{aligned} \mathrm{u}(x,y) &= a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 \\ \mathrm{v}(x,y) &= b_0 + b_1 x + b_2 y + b_3 x^2 + b_4 xy + b_5 y^2 \end{aligned} \qquad (6.1)$$

Although neither the Finite Element quadratic element nor the 2nd order DDA block developed here use this form, it may be demonstrated that the functions used by both methods are equivalent to Equation 6.1.

6.1 The DDA Formulation of the 2nd Order Element.

6.1.1 Element degrees of freedom

It was an assumption of the first order block that strains were constant throughout the area of the block. The second order displacement function allows strains to vary linearly across the block. The deformation parameters chosen for each block are therefore:

- the block rigid body motions at the centroid $(\mathrm{u}_0 \quad \mathrm{v}_0 \quad \mathrm{r}_0)$,
- the strains measured at the centroid $(\varepsilon_x^0 \quad \varepsilon_y^0 \quad \gamma_{xy}^0)$, and
- the rates of variation of the strains, $(\varepsilon_{x,x} \quad \varepsilon_{x,y} \quad \varepsilon_{y,x} \quad \varepsilon_{y,y} \quad \gamma^*_{xy,x} \quad \gamma^*_{xy,y})$.

The two terms $\gamma^*_{xy,x}$ and $\gamma^*_{xy,y}$ are marked with asterixes because, as will be shown in Equation 6.3, the shear strain is also affected by other degrees of freedom. These two terms mark the variational strain caused by the pure shear component.

The element displacement function is written as:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{1}{2}\bar{y} & \frac{1}{2}\bar{x}^2 & \bar{x}\bar{y} & 0 & 0 & 0 & \frac{1}{2}\bar{y}^2 \\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{1}{2}\bar{x} & 0 & 0 & \bar{x}\bar{y} & \frac{1}{2}\bar{y}^2 & \frac{1}{2}\bar{x}^2 & 0 \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \\ r_0 \\ \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \\ \varepsilon_{x,x} \\ \varepsilon_{x,y} \\ \varepsilon_{y,x} \\ \varepsilon_{y,y} \\ \gamma^*_{xy,x} \\ \gamma^*_{xy,y} \end{pmatrix}$$

or

$$\begin{pmatrix} u \\ v \end{pmatrix} = \underline{\underline{T}}^i(x,y)\underline{d}^i \qquad (6.2)$$

where

$$\bar{x} = x - x_0$$
$$\bar{y} = y - y_0$$

and (x_0, y_0) is the centroid of the block.

From this displacement function, it follows that

$$\varepsilon_x(x,y) = \frac{\partial u}{\partial x} = \varepsilon_x^0 + \bar{x}\varepsilon_{x,x} + \bar{y}\varepsilon_{x,y}$$
$$\varepsilon_y(x,y) = \frac{\partial v}{\partial y} = \varepsilon_y^0 + \bar{x}\varepsilon_{y,x} + \bar{y}\varepsilon_{y,y}$$
$$\gamma_{xy}(x,y) = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \gamma_{xy}^0 + \bar{x}\left(\gamma^*_{xy,x} + \varepsilon_{x,y}\right) + \bar{y}\left(\gamma^*_{xy,y} + \varepsilon_{y,x}\right)$$

or

$$\begin{pmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{pmatrix}=\begin{bmatrix}0&0&0&1&0&0&\bar{x}&\bar{y}&0&0&0&0\\ 0&0&0&0&1&0&0&0&\bar{x}&\bar{y}&0&0\\ 0&0&0&0&0&1&0&\bar{x}&\bar{y}&0&\bar{x}&\bar{y}\end{bmatrix}\underline{d}^i$$

$$\underline{\varepsilon}^i(x,y)=\underline{\underline{B}}^i(x,y)\underline{d}^i \quad . \tag{6.3}$$

6.1.2 Equilibrium of a single block

The method of determining the internal equilibrium equations is similar to that used for the 1st order blocks. The most important difference is that the stresses and strains are no longer constant, and this alters the derivation of the internal strain energy.

As before, the internal strain energy of the deformed block is evaluated as

$$\Pi_e=\int\int\frac{1}{2}\left(\sigma_x\varepsilon_x+\sigma_y\varepsilon_y+\tau_{xy}\gamma_{xy}\right)dx.dy$$

$$\Pi_e=\frac{1}{2}\int\int\begin{pmatrix}\varepsilon_x & \varepsilon_y & \gamma_{xy}\end{pmatrix}\begin{pmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{pmatrix}.dx.dy \quad . \tag{6.4}$$

Define, for plane stress conditions,

$$\begin{pmatrix}\sigma_x\\ \sigma_y\\ \tau_{xy}\end{pmatrix}=\frac{E}{1-\nu^2}\begin{bmatrix}1&\nu&0\\ \nu&1&0\\ 0&0&\frac{1-\nu}{2}\end{bmatrix}\begin{pmatrix}\varepsilon_x\\ \varepsilon_y\\ \gamma_{xy}\end{pmatrix}$$

$$\underline{\sigma}^i(x,y)=\underline{\underline{D}}\ \underline{\varepsilon}^i(x,y) \quad . \tag{6.5}$$

Substituting equations 6.5 and 6.3 into equation 6.4,

$$\Pi_e=\frac{1}{2}\int\int\underline{\varepsilon}^i(x,y)^T\underline{\underline{D}}\ \underline{\varepsilon}^i(x,y).dx.dy$$

$$\Pi_e=\frac{1}{2}\int\int\underline{d}^{iT}\underline{\underline{B}}^i(x,y)^T\underline{\underline{D}}\ \underline{\underline{B}}^i(x,y)\underline{d}^i.dx.dy \quad . \tag{6.6}$$

Minimisation of the strain potential energy with respect to the block degrees of freedom yields

$$K_{rs}^{ii} = \frac{\partial^2 \Pi_e}{\partial d_r^i . \partial d_s^i} = \frac{\partial^2}{\partial d_r^i . \partial d_s^i}\left(\underline{d}^{iT} \frac{1}{2} \int\int \underline{\underline{B}}^i(x,y)^T \underline{\underline{D}}\ \underline{\underline{B}}^i(x,y).dx.dy.\underline{d}^i \right)$$

$$= \int\int B_{tr}^i(x,y) D_{tu} B_{us}^i(x,y).dx.dy$$

$$\int\int \underline{\underline{B}}^i(x,y)^T \underline{\underline{D}}\ \underline{\underline{B}}^i(x,y).dx.dy \Rightarrow \underline{\underline{K}}^{ii} \quad . \qquad (6.7)$$

This is evaluated directly:

$$\underline{\underline{K}}^{ii} = \frac{E}{1-\nu^2} \int\int \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & \nu & 0 & \bar{x} & \bar{y} & \nu\bar{x} & \nu\bar{y} & 0 & 0 \\
0 & 0 & 0 & \nu & 1 & 0 & \nu\bar{x} & \nu\bar{y} & \bar{x} & \bar{y} & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2} & 0 & \frac{1-\nu}{2}\bar{x} & \frac{1-\nu}{2}\bar{y} & 0 & \frac{1-\nu}{2}\bar{x} & \frac{1-\nu}{2}\bar{y} \\
0 & 0 & 0 & \bar{x} & \nu\bar{x} & 0 & \bar{x}^2 & \bar{x}\bar{y} & \nu\bar{x}^2 & \nu\bar{x}\bar{y} & 0 & 0 \\
0 & 0 & 0 & \bar{y} & \nu\bar{y} & \frac{1-\nu}{2}\bar{x} & \bar{x}\bar{y} & \bar{y}^2 + \frac{1-\nu}{2}\bar{x}^2 & \frac{1+\nu}{2}\bar{x}\bar{y} & \nu\bar{y}^2 & \frac{1-\nu}{2}\bar{x}^2 & \frac{1-\nu}{2}\bar{x}\bar{y} \\
0 & 0 & 0 & \nu\bar{x} & \bar{x} & \frac{1-\nu}{2}\bar{y} & \nu\bar{x}^2 & \frac{1+\nu}{2}\bar{x}\bar{y} & \bar{x}^2 + \frac{1-\nu}{2}\bar{y}^2 & \bar{x}\bar{y} & \frac{1-\nu}{2}\bar{x}\bar{y} & \frac{1-\nu}{2}\bar{y}^2 \\
0 & 0 & 0 & \nu\bar{y} & \bar{y} & 0 & \nu\bar{x}\bar{y} & \nu\bar{y}^2 & \bar{x}\bar{y} & \nu\bar{y}^2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2}\bar{x} & 0 & \frac{1-\nu}{2}\bar{x}^2 & \frac{1-\nu}{2}\bar{x}\bar{y} & 0 & \frac{1-\nu}{2}\bar{x}^2 & \frac{1-\nu}{2}\bar{x}\bar{y} \\
0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2}\bar{y} & 0 & \frac{1-\nu}{2}\bar{x}\bar{y} & \frac{1-\nu}{2}\bar{y}^2 & 0 & \frac{1-\nu}{2}\bar{x}\bar{y} & \frac{1-\nu}{2}\bar{y}^2
\end{bmatrix} .dx.dy$$

$$\underline{\underline{K}}^{ii} = \frac{E}{1-\nu^2} \begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & S^i & \nu S^i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & \nu S^i & S^i & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2}S^i & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & S_{xx}^i & S_{xy}^i & \nu S_{xx}^i & \nu S_{xy}^i & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & S_{xy}^i & \frac{1-\nu}{2}S_{xx}^i + S_{yy}^i & \frac{1+\nu}{2}S_{xy}^i & \nu S_{yy}^i & \frac{1-\nu}{2}S_{xx}^i & \frac{1-\nu}{2}S_{xy}^i \\
0 & 0 & 0 & 0 & 0 & 0 & \nu S_{xx}^i & \frac{1+\nu}{2}S_{xy}^i & S_{xx}^i + \frac{1-\nu}{2}S_{yy}^i & S_{xy}^i & \frac{1-\nu}{2}S_{xy}^i & \frac{1-\nu}{2}S_{yy}^i \\
0 & 0 & 0 & 0 & 0 & 0 & \nu S_{xy}^i & \nu S_{yy}^i & S_{xy}^i & S_{yy}^i & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2}S_{xx}^i & \frac{1-\nu}{2}S_{xy}^i & 0 & \frac{1-\nu}{2}S_{xx}^i & \frac{1-\nu}{2}S_{xy}^i \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2}S_{xy}^i & \frac{1-\nu}{2}S_{yy}^i & 0 & \frac{1-\nu}{2}S_{xy}^i & \frac{1-\nu}{2}S_{yy}^i
\end{bmatrix}$$

. (6.8)

S^i, S_{xx}^i, S_{xy}^i, S_{yy}^i are defined in Section 3.3.6, and they are determined by simplex integration. It should be noted that the simplexes are calculated on the assumption that the edges between nodes are straight. This is not true in the case of 2nd order blocks; however materials such as rock will not accept large deformations before

failing, and therefore the curvature of the block boundary segments is small, and does not introduce significant errors.

The method of formulation of the energy minimisation equations for the initial stress, point and volume loading, forces of inertia and assigned displacements is the same as that for 1st order blocks. Because the $\underline{\underline{T}}^i(x, y)$ matrix is now a 2 x 12 array, the formulation results in a 12 x 12 block stiffness matrix, and a 12 x 1 force vector.

6.1.3 Block system interaction

Interaction between blocks is again due to rock bolts and contact. The formulation for rock bolt interaction is similar to that for 1st order elements, except that here the 2nd order $\underline{\underline{T}}^i(x, y)$ matrix is used.

Contact penetration distances are calculated on the assumption that the block edges between vertices remain straight. This is necessary in order to simplify the calculations. As with simplex integration, the assumption is justified on the basis that curvature of block boundary segments is small.

With this approximation, the contact formulation is the same as that for the 1st order elements.

6.2 The Meshing Procedure for 2nd Order Elements

The compatibility condition of the Finite Element method stipulates that while the strains of elements may be discontinuous at common element boundaries, the displacements must be continuous. The 2nd order element has a quadratic displacement function, so that the displaced shape of a block edge may be uniquely described by the displacement of three points along that edge. Thus, if two neighbouring elements share three nodes along their common edge, and compatibility exists at these three nodes, then compatibility will be satisfied at all points along the shared boundary.

The Finite Element equivalent of the 2nd order DDA block is the C^0-quadratic triangle shown in Figure 6.1.

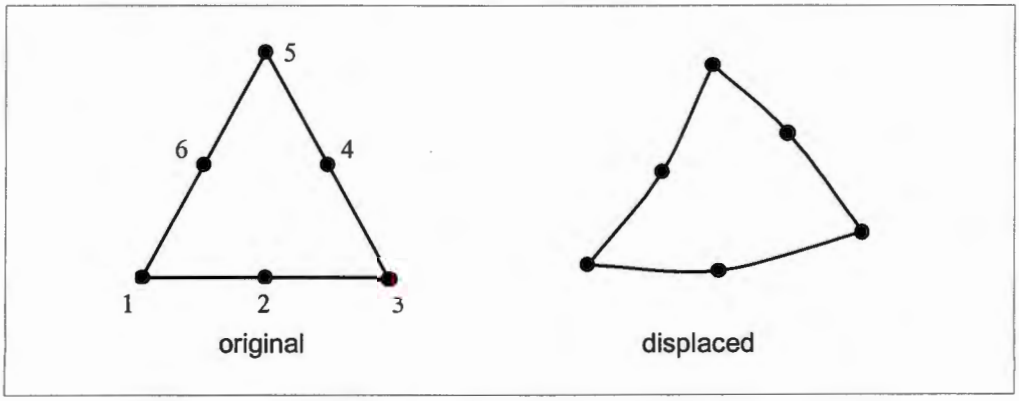


Figure 6.1: A C^0-quadratic triangular element.

The 2nd order DDA element and the C^0-quadratic triangular element both have twelve degrees of freedom. In order to render the block equilibrium equations in terms of the nodal displacements, it is again necessary to formulate a matrix which relates the two different sets of degrees of freedom. The method is the same as that used in Chapter 5:

$$\begin{pmatrix} u_1 \\ v_1 \\ u_2 \\ v_2 \\ \vdots \\ \vdots \\ u_6 \\ v_6 \end{pmatrix} = \begin{bmatrix} \left[\underline{\underline{T}}^i(x_1, y_1) \right] \\ \hline \left[\underline{\underline{T}}^i(x_2, y_2) \right] \\ \hline \vdots\vdots\vdots \\ \vdots\vdots\vdots \\ \left[\underline{\underline{T}}^i(x_6, y_6) \right] \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \\ r_0 \\ \varepsilon_x^0 \\ \vdots \\ \vdots \\ \gamma^*_{xy,x} \\ \gamma^*_{xy,y} \end{pmatrix}$$

$$\underline{u}^i = \underline{\underline{R}}^i \underline{d}^i \tag{6.9}$$

and

$$\underline{d}^i = \left[\underline{\underline{R}}^i \right]^{-1} \underline{u}^i = \underline{\underline{Q}}^i \underline{u}^i \tag{6.10}$$

where $\underline{\underline{R}}^i$ and $\underline{\underline{Q}}^i$ are 12 x 12 matrices.

The procedure for meshing is then the same as for the 1st order elements.

6.3 Performance of 2nd Order Element Meshes

In order to test the performance of the 2nd order element, a further code was written using Mathcad Plus 6.0 [1995]. This code was designed to test the performance of a single block, sub-meshed using 2nd order elements, without interaction with other blocks. The code is printed in Appendix C.

Four tests were performed on the 2nd order element, using this code. The patch test is a preliminary test to ensure that the element conforms to the completeness criterion. The Cook cantilever test, and the uniaxial compression problem were analysed using 2nd order elements as a comparison with the 1st order element tests of Chapter 5. Finally, tests were performed on a beam in bending. An important motivation for developing the more complicated 2nd order element is the poor performance of the 1st order element in bending. Therefore, it is necessary to obtain a measure of the accuracy with which bending is modelled by the 2nd order element.

These tests are described in this section.

6.3.1 Convergence criteria

In order to guarantee convergence, the element must conform to the compatibility and completeness criteria (see Section 5.8.1). The compatibility criterion has already been discussed in Section 6.2. In order to test the completeness criterion, a patch test was performed on the element. Details of the test may be found in Appendix C.

The 2nd order element passed the patch test.

6.3.2 The Cook cantilever test

The Cook cantilever test was described in Section 5.7.3. Various meshes of increasing refinement were tested to establish convergence, using 1st and 2nd order elements. These meshes are depicted in Figure 6.2. Table 6.1 compares the results of the different analyses. The program code was unable to analyse meshes (e) to (g) using 2nd order elements due to memory constraints.

In Figure 6.3, the results are plotted against the total number of degrees of freedom of the meshes. It can be seen that the 2nd order element converges toward the 'best known value' considerably more rapidly than the 1st order element.

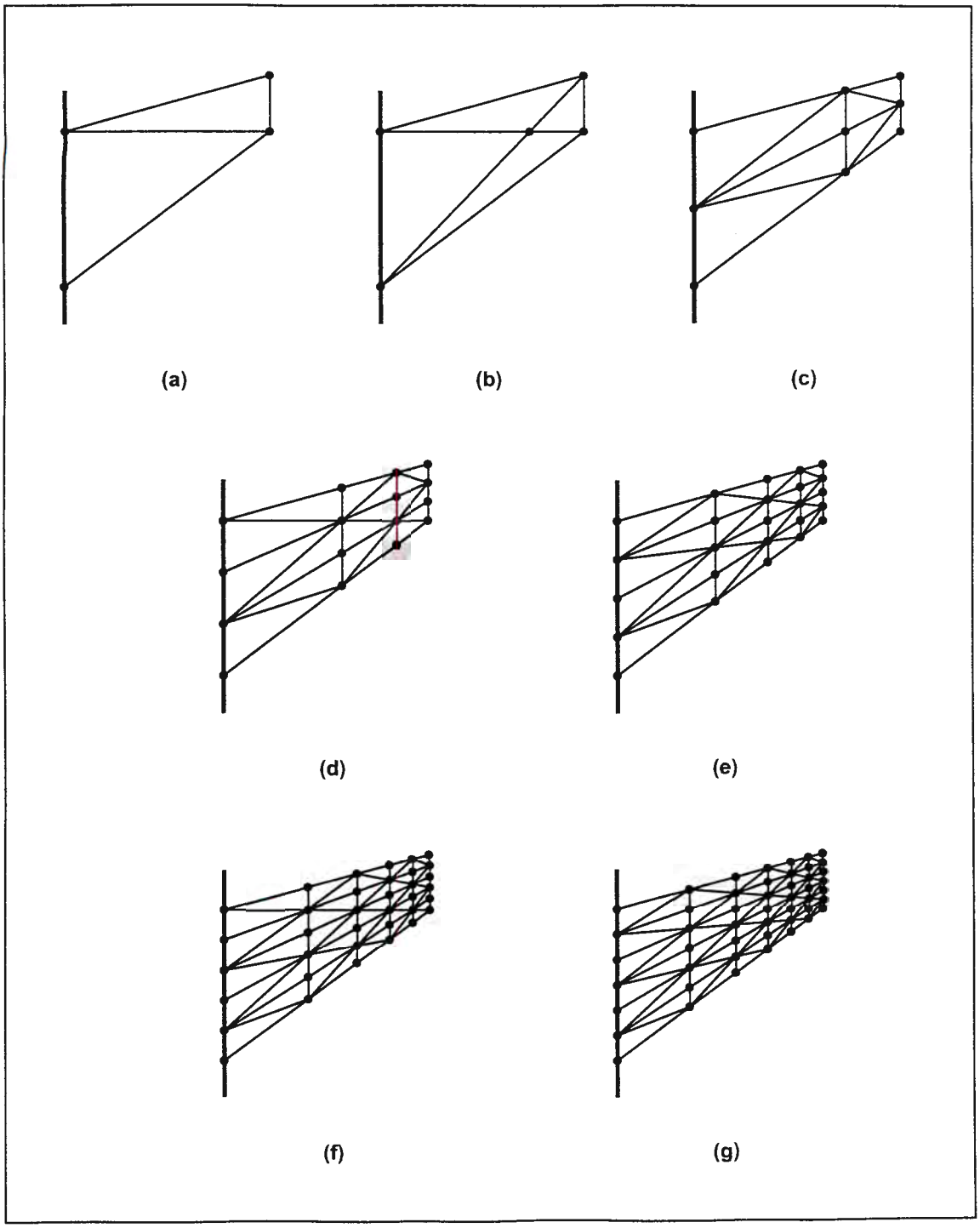


Figure 6.2: Mesh geometries for the Cook cantilever test.

Mesh	No. of elements	1st Order Elements			2nd Order Elements		
		No. of nodes	No. of d.o.f.	v_B / v_{B0}	No. of nodes	No. of d.o.f.	v_B / v_{B0}
(a)	2	4	8	0.251	9	18	0.753
(b)	4	5	10	0.262	13	26	0.784
(c)	8	9	18	0.302	25	50	0.884
(d)	18	16	32	0.456	49	98	0.949
(e)	32	25	50	0.529	-	-	-
(f)	50	36	72	0.650	-	-	-
(g)	72	49	98	0.693	-	-	-

Table 6.1: Comparison of element performance in the Cook cantilever test.

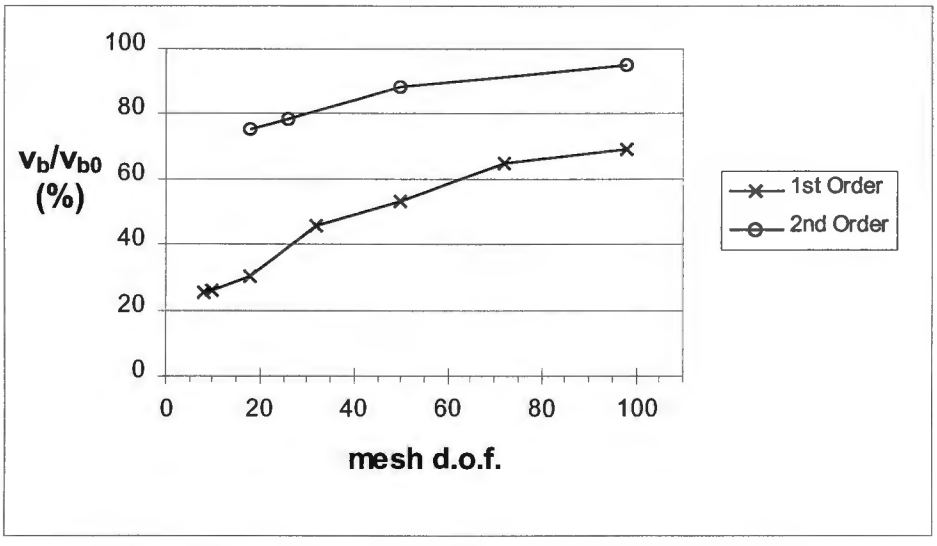


Figure 6.3: Convergence in the Cook cantilever test.

6.3.3 Uniaxial compression test

A uniaxial compression test was modelled using 2nd order elements. The geometry and loading, and the number and distribution of nodes in the mesh was kept the same as for the 1st order element mesh of Section 5.8.4.

The number of nodal degrees of freedom in the problem is again 76. The mesh consists of 14 2nd order elements, so that there were 168 original sub-block deformation variables. It can be shown that in a regular mesh of 2nd order elements, as the number of elements in the mesh approaches infinity the ratio of the number of nodal displacement degrees of freedom to the number of sub-block deformation parameters approaches 1 : 3.

Contours of the calculated Von Mises stresses in the mesh are plotted in Figures 6.4 and 6.5. Figure 6.4 is produced by the DDA meshing formulation, while Figure 6.5 is produced by the Abaqus Finite Element program [1996] using 6-noded C^0-quadratic elements. It can be seen that the two results are similar within the range of rounding errors.

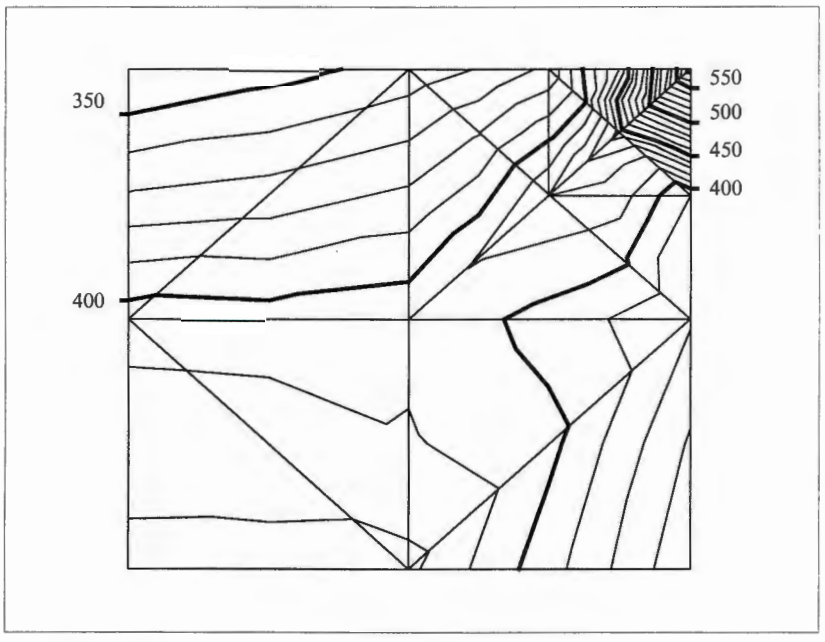


Figure 6.4: DDA solution: Contour plot of Von Mises stresses (MPa) in the uniaxial compression test.

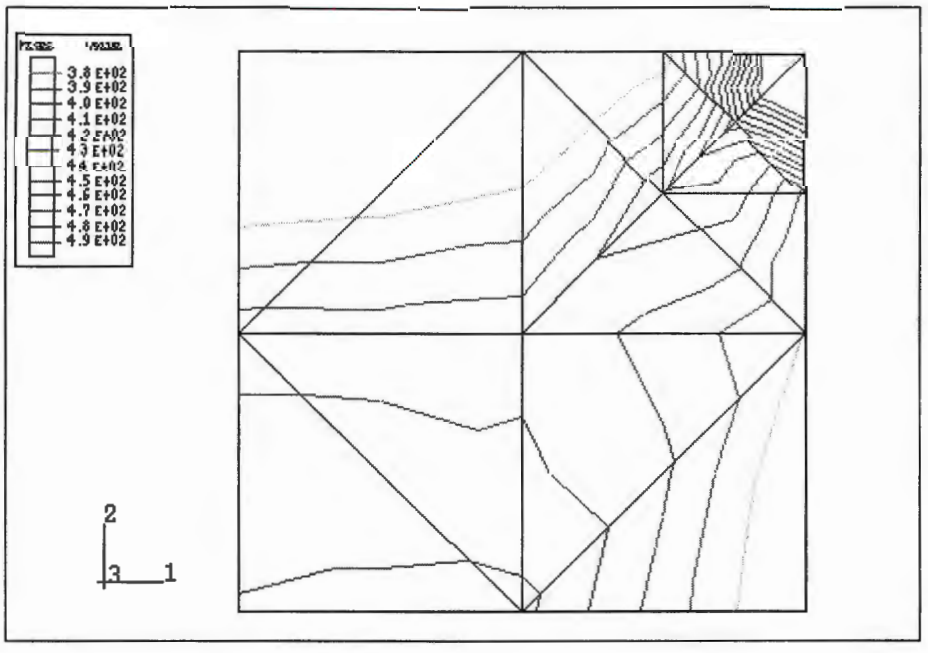


Figure 6.5: FEM solution: Contour plot of Von Mises stresses in the uniaxial compression test. Contours are in the range 380 – 490 MPa.

6.3.4 Prismatic cantilever in bending

Figure 6.6 shows three simple analyses of a cantilever under loading. The cantilever is loaded by a force couple at its free end to induce pure bending along the length of the beam. Mesh (a) is the simplest manner in which a cantilever may be modelled using triangular elements. The elements in mesh (b) are arranged so as to investigate the effect of element distortion upon the solution. Mesh (c) is oriented at an angle of approximately 0.644 radians to the x-axis in order to confirm that element orientation does not have any effect on the solution.

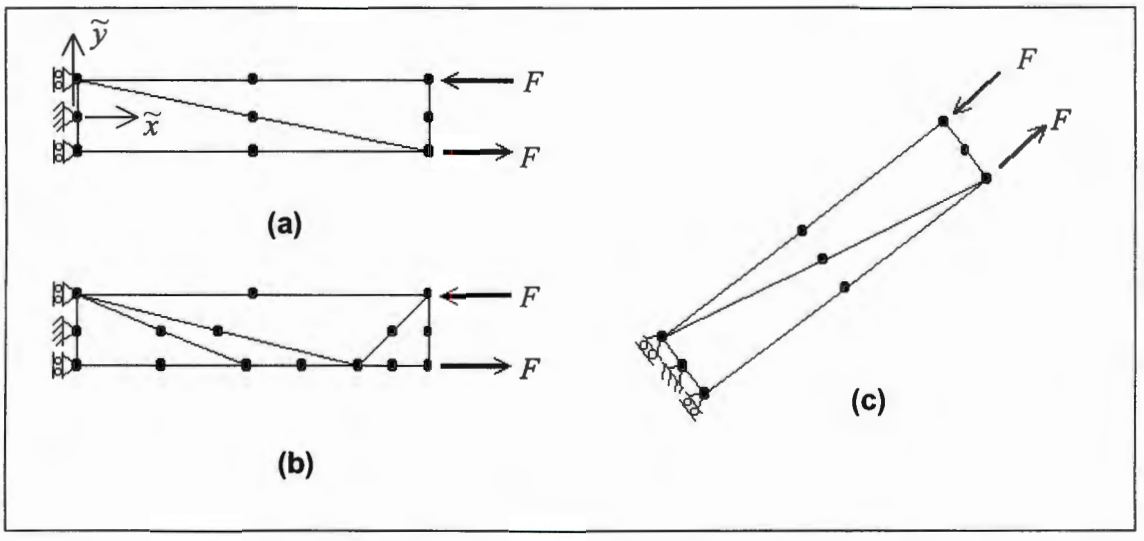


Figure 6.6: Mesh geometries for a cantilever in bending.

The stresses and deflections in a cantilever under pure bending can be obtained using small deflection theory. Using the local axis system marked in Figure 6.6(a), the strains in the beam are:

$$\varepsilon_{\tilde{x}\tilde{x}} = \frac{M\tilde{y}}{EI} = \frac{12F\tilde{y}}{Eh^2}$$

$$\varepsilon_{\tilde{y}\tilde{y}} = \nu\varepsilon_{\tilde{x}\tilde{x}}$$

$$\gamma_{\tilde{x}\tilde{y}} = 0$$

where h is the height of the beam in the $\tilde{y}$ direction. Similarly, the deflections (u, v) at any point along the beam centre-line are:

$$\mathrm{u} = \frac{M\tilde{x}^2}{2EI} = \frac{6F\tilde{x}^2}{Eh^2}$$

$$\mathrm{v} = 0$$

The results obtained for all three meshes are in exact conformity with the theoretical solution.

Chapter 7

POTENTIAL EXTENSIONS TO THE MESHING METHOD

In order to fully exploit the advantages of a meshing capability within the DDA method, some further developments are desirable. Three possible developments are discussed in this chapter:

- Further element types. Two elements have been investigated and applied to DDA in the preceding chapters. The possibility of developing other elements, for ease of meshing and improved mesh performance, is discussed.
- Material non-linearity. Most engineering materials have a non-linear response to loading. The question of incorporating more accurate material models than the linear-elastic model into the DDA method needs to be addressed.
- Fracture. On the one hand, the more accurate stress determination provided by meshing allows the prediction of fracture in a body, while on the other hand, DDA is designed to investigate the interaction of separate bodies in a discontinuous system. The DDA method therefore has excellent potential for predicting the onset of fracture and failure of engineering materials, and thereafter modelling the consequences of that failure. It would be very useful to incorporate a fracture mechanism whereby bodies can fracture into a number of separate bodies, with the analysis automatically continuing under the new configuration.

7.1 Further Element Types

Much of the discussion in this section is derived from "Finite Element Analysis, from Concepts to Applications", by D.S. Burnett [1987].

In considering two-dimensional element displacement functions, it is useful to display the polynomial terms in a triangular form, known as Pascal's triangle, the first 7 rows of which are depicted in Figure 7.1.

All terms have the form $x^r y^s$ and a term is of degree p if $r+s=p$.

A polynomial function is said to be complete to degree p if it contains all terms up to and including those of degree p.

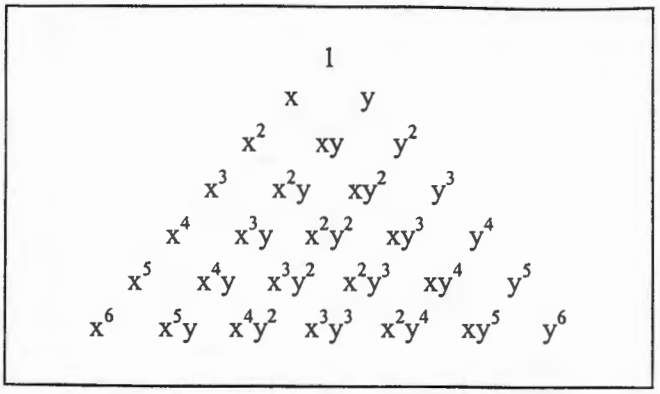


Figure 7.1: Pascal's triangle.

It can be seen that the 1st order element of Chapter 5 was complete to degree 1, and the 2nd order element of Chapter 6 was complete to degree 2. The rate of convergence of an element depends on the degree of the completeness of the polynomial function. If an element displacement function contains higher-order terms, but is not complete in these degrees, the solution may be improved at certain points in the element, but convergence is not guaranteed to be greater than that expected for the degree of completeness.

There are many conceivable shapes for elements. However, in two dimensional Finite Element analyses, triangular and quadrilateral elements are used almost exclusively because of the convenience of meshing and formulation.

7.1.1 Triangular Elements

Table 7.1 shows the first four in the series of C^0-triangular elements, and the terms that are included in their polynomial displacement functions. It can be seen that each displacement function displays *geometric isotropy*, meaning that it is complete to a particular degree, and contains no terms of higher degree.

The first two elements in this series have been discussed already. The higher-order elements could also be incorporated into the DDA method.

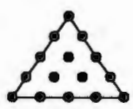
C^0-linear	C^0-quadratic	C^0-cubic	C^0-quartic
			
1 $x \quad y \quad p=1$	1 $x \quad y$ $x^2 \quad xy \quad y^2 \quad p=2$	1 $x \quad y$ $x^2 \quad xy \quad y^2$ $x^3 \quad x^2y \quad xy^2 \quad y^3 \quad p=3$	1 $x \quad y$ $x^2 \quad xy \quad y^2$ $x^3 \quad x^2y \quad xy^2 \quad y^3$ $x^4 \quad x^3y \quad x^2y^2 \quad xy^3 \quad y^4 \quad p=4$

Table 7.1: The first four elements in the C^0-triangle series, and their displacement function terms.

7.1.2 Quadrilateral Elements

There are two common families of quadrilateral elements in the Finite Element method. These are the Lagrange series and the serendipity series. Examples of these elements, and the terms that make up their displacement functions, are depicted in Tables 7.2 and 7.3 respectively. It may be seen that the first element is common to both series, and this element will be discussed in more detail.

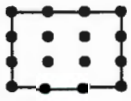
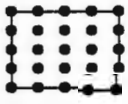
Bilinear	Biquadratic	Bicubic	Biquartic
			
1 $x \quad y \quad p=1$ xy	1 $x \quad y$ $x^2 \quad xy \quad y^2 \quad p=2$ $x^2y \quad xy^2$ x^2y^2	1 $x \quad y$ $x^2 \quad xy \quad y^2$ $x^3 \quad x^2y \quad xy^2 \quad y^3 \quad p=3$ $x^3y \quad x^2y^2 \quad xy^3$ $x^3y^2 \quad x^2y^3$ x^3y^3	1 $x \quad y$ $x^2 \quad xy \quad y^2$ $x^3 \quad x^2y \quad xy^2 \quad y^3$ $x^4 \quad x^3y \quad x^2y^2 \quad xy^3 \quad y^4 \quad p=4$ $x^4y \quad x^3y^2 \quad x^2y^3 \quad xy^4$ $x^4y^2 \quad x^3y^3 \quad x^2y^4$ $x^4y^3 \quad x^3y^4$ x^4y^4

Table 7.2: The first four elements in the 2D-Lagrange series, and their displacement function terms.

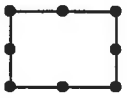
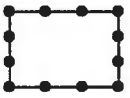
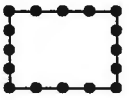
Linear	Quadratic	Cubic	Quartic
			
1 x y p=1 - xy -	1 x y x^2 xy y^2 p=2 - x^2y xy^2 -	1 x y x^2 xy y^2 x^3 x^2y xy^2 y^3 p=3 - x^3y - xy^3 -	1 x y x^2 xy y^2 x^3 x^2y xy^2 y^3 x^4 x^3y x^2y^2 xy^3 y^4 p=4 - x^4y - - xy^4 -

Table 7.3: The first four elements in the 2D-serendipity series, and their displacement function terms.

The bi-linear quadrilateral element has 8 degrees of freedom. From the foregoing discussion, it would appear that the equivalent DDA element would have the displacement function:

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{\bar{y}}{2} & \overline{xy} & 0 \\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{\bar{x}}{2} & 0 & \overline{xy} \end{bmatrix} \begin{pmatrix} u_0 \\ v_0 \\ r_0 \\ \varepsilon_x^0 \\ \varepsilon_y^0 \\ \gamma_{xy}^0 \\ \varepsilon_{x,y} \\ \varepsilon_{y,x} \end{pmatrix}$$

This formulation displays geometric anisotropy, however, and this has important consequences in the DDA method. Because the displacement function contains the term $\overline{xy}$ the displaced shape of an edge of an element will in general be curved. It cannot be guaranteed that the edge of a neighbouring element in a mesh will curve in a similar manner, and so the continuity of the mesh is lost.

In the Finite Element formulation, the element displacement is defined in terms of a local, non-orthogonal co-ordinate system, as illustrated in Figure 7.2. The axes of this coordinate system are ξ and η, and it can be seen that along each element boundary either ξ or η is constant, so that the term $\xi\eta$ varies linearly along the boundaries.

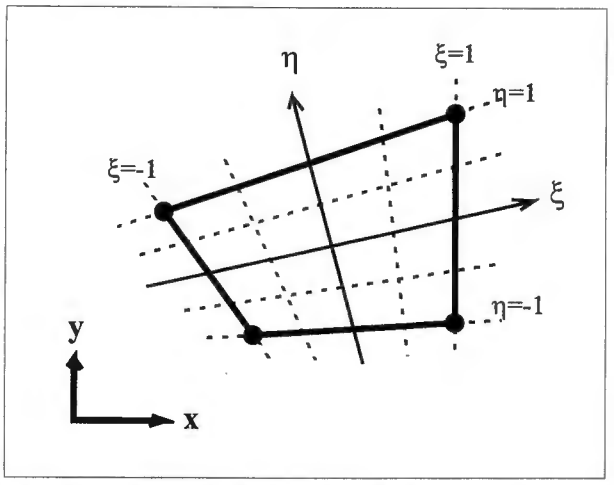


Figure 7.2: The element co-ordinate system used in FEM.

It would be conceivable to incorporate the equivalent of shape functions into the DDA method. The displacement function would become

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{bmatrix} 1 & 0 & -\bar{y} & \bar{x} & 0 & \frac{\bar{y}}{2} & \xi(\bar{x},\bar{y})\eta(\bar{x},\bar{y}) & 0 \\ 0 & 1 & \bar{x} & 0 & \bar{y} & \frac{\bar{x}}{2} & 0 & \xi(\bar{x},\bar{y})\eta(\bar{x},\bar{y}) \end{bmatrix} \left(\underline{d}^i\right)$$

where

$$\xi(\bar{x},\bar{y}) = a_0 + a_1\bar{x} + a_2\bar{y} + a_3\bar{x}\bar{y}$$
$$\eta(\bar{x},\bar{y}) = b_0 + b_1\bar{x} + b_2\bar{y} + b_3\bar{x}\bar{y}$$

This formulation would then contain higher order terms, however, and would become considerably more complex as a result. For the higher order elements in the series, the local co-ordinate systems would have curved axes, further complicating the calculations.

It appears that the method proposed in this report is only suitable for blocks and elements that display geometric isotropy. In effect, this limits element types to those of the C^0-triangle series.

7.2 Material Non-Linearity

Incorporating material non-linearity is a possibility with the DDA meshing method. The procedure for this would be similar to that used in the Finite Element method. A stress update procedure must be implemented, and this would depend on the material model. An iterative solution method is then used, solving for the residual forces until convergence is obtained. The matrix $\underline{\underline{E}}$ of Equation 3.9 or $\underline{\underline{D}}$ of Equation 6.5 is replaced

by a tangent modulus which is dependent on the state of stress and load history of the element.

The formulation is convenient to implement, because only the term for elastic strain in the total potential energy equations will be altered by non-linearity.

It may be noted that the artificial joint method of Ke and Goodman [1994] does not allow non-linear formulations because the blocks do not directly share degrees of freedom, so that the residual forces cannot be established.

7.3 Fracture

There are two possibilities for incorporating fracture into DDA, where blocks are meshed using first-order elements. Both procedures would require a failure criterion such as the Mohr-Coulomb or Von Mises criteria.

The simplest method of introducing fracture is to allow the block to fracture only along element boundaries. The internal element forces at each corner node would be derived from the system solution, using the equation

$$\underline{\underline{F}}^{i}_{\text{int}} = S^{i}\underline{\underline{Q}}^{T}\underline{\sigma}^{i} \tag{7.1}$$

where

$$\underline{\underline{F}}^{i}_{\text{int}} = \begin{pmatrix} F_{1x} & F_{1y} & F_{2x} & F_{2y} & F_{3x} & F_{3y} \end{pmatrix}^{T}$$

and

$$\underline{\sigma}^{i} = \begin{pmatrix} 0 & 0 & 0 & \sigma_{x} & \sigma_{y} & \tau_{xy} \end{pmatrix}^{T} \quad .$$

It can be arranged that, if the failure criterion is exceeded at any node, then the mesh is made discontinuous at that point by splitting the node into two. Failure would be progressive in this way. Contact determination would need to be carried out along the fracture as the analysis proceeded, to ensure that overlapping of the blocks did not occur.

This method would be anisotropic, with failure being strongly dependent on the orientation of the element boundaries.

A more sophisticated approach would be to consider the stresses at the element centres, instead of the boundaries. If it is determined that failure has occurred, then the element splits in a direction determined by the failure criterion. The fracture passes

through one of the nodes, with the other end forming a further node on the opposite edge of the element. The element adjacent to this edge is divided into two elements, to maintain the corner-to-corner meshing required for continuity.

In the following increment, further stress determination would determine whether the fracture would extend into neighbouring elements.

Figure 7.3 shows this scheme diagrammatically. In (a), a portion of the original mesh is depicted. If it is found that element 1 has reached failure then it splits, and element 2 is re-meshed, as shown in (b), although the new boundary in element 2 is not a discontinuity. Failure in subsequent time increments may occur along the dotted lines indicated.

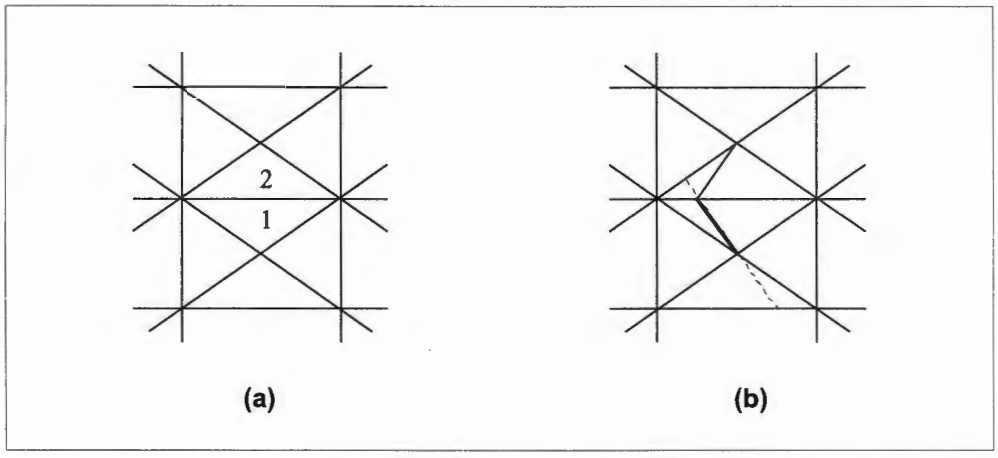


Figure 7.3: Fracture within a mesh.

Chapter 8

CONCLUSIONS

A blocky rock mass is an extremely complicated physical system. If it were attempted to model the system at its full level of complexity, a numerical simulation of a rock mass would fail due to computational cost. Therefore, all discrete element methods provide approximate solutions, making simplifying assumptions in order to arrive at a compromise between computational efficiency and solution accuracy.

While the mathematical formulation of the DDA method is very efficient, the quality of the results obtained is dependent on certain control parameters, notably the time-step interval and the contact penalty value. Because of this, these values should be carefully chosen in order to obtain accurate results. The DDA software is able to control these parameters automatically during the course of an analysis, but the quality of this control is poor, particularly in the case of the penalty value. The manner in which the penalty value is controlled by the DDA program needs to be revised. However, until this is done it is recommended that the user specify a fixed value for the penalty value, and that a series of analyses are performed, comparing the results obtained using different values of the control parameters, in order to ensure that results are consistent.

With correct control of these parameters, the DDA method can yield accurate results, both in static and dynamic analyses. However, modelling of dynamic contact between blocks is inaccurate if relative velocities are high, and the DDA method is not suitable for this type of analysis.

One of the simplifying assumptions that the DDA method makes is that of constant strains within blocks. This assumption is generally reasonable, but situations often arise where a more accurate description of strains within particular blocks or structural elements is needed in order to obtain meaningful results.

The ability to sub-divide individual blocks into Finite Element meshes within the global system provides the user with a convenient method of selectively controlling the accuracy of strain determination within a solution, and hence the computational cost of the analysis.

The DDA method and the Finite Element method both use the principle of stationary potential energy to obtain equilibrium within the system. There are close parallels

between the solution equations of the two analysis techniques, therefore, despite the fact that the two methods are designed to solve very different problems.

The sub-meshing method described in this report takes advantage of the similarity of the two methods by initially treating both blocks and sub-block elements as DDA blocks. The standard DDA code may then be used to develop the solution equations for the system, and there is no need to introduce a Finite Element code into the program. Only once the solution equations have been formulated are they selectively transformed into FEM format before solution.

The end result is a mixed formulation, with DDA blocks interacting with FEM meshes within a single system. The user can therefore simulate a rock mass by using any desired combination of 1st and 2nd order blocks and element meshes.

This method is more efficient than the artificial joint concept of Ke and Goodman [1994] because mesh displacements are described by the displacements of the nodes, rather than the sub-block deformation parameters. Although the two descriptions are equivalent to each other, there is a high degree of inter-dependency between the degrees of freedom of the latter. In the case of 1st order elements, the formulation in terms of block deformations may contain up to six times as many variables as the formulation using nodal displacements. Similarly, for 2nd order elements, there are up to three times as many unknowns if the displaced shape is described in terms of the block deformations, rather than the nodal displacements.

In a DDA analysis, the stiffness matrix must be formulated and inverted many times, and this represents a large part of the computational expense of the analysis. Any reduction in the size of the stiffness matrix therefore lessens the time taken to solve a problem.

The sub-meshing method described here is not as comprehensive as the mixed formulation of Chang [1994]. Chang's method introduces a complete Finite Element code, and therefore is able to make full use of the techniques developed for the Finite Element method. In particular, the various quadrilateral elements used in the Finite Element method can be used in analyses. These elements have been found to be more efficient than triangular elements. The sub-meshing method is considerably simpler than Chang's method, however. Whereas the mixed formulation is in effect two separate methods working in parallel, the sub-meshing method works within the DDA method and its implementation does not require extensive revision of the DDA code.

1st order element meshes are not efficient in modelling bending. However, the 2nd order element is able to model bending to a high degree of accuracy using relatively simple meshes. In the sub-meshing method, therefore, most modes of deformation can be modelled with a reasonable degree of accuracy and efficiency using just these two element types.

The sub-meshing method can model non-linear material behaviour in the same way that it is modelled in Finite Element analyses. Fracture of blocks can also be conveniently introduced.

In general, the method proposed here greatly enhances the scope of the DDA method in modelling rock masses, while remaining efficient and simple to implement.

List of References

ABAQUS Version 5.6 (1996) © Hibbitt, Karlsson and Sorensen, Inc., 1080 Main St., Pawtucket, RI.

AMADEI, B., CHIHSEN, L. and DWYER, J. (1996) Recent extensions to the DDA method. *Proceedings, First International Forum on DDA and simulations of discontinuous media*, Berkeley, California. pp 1-30.

BLAKE, W. (1969) Finite Element model is excellent pit tool. *Mining Engineering*, A.I.M.E., Vol. 21, No. 8, pp 79-80.

BARLOW, J. (1976) Optimal stress locations in Finite Element models. *International Journal for Numerical Methods in Engineering*, Vol 10, pp 243-251.

BURNETT, D.S. (1987) Finite Element analysis from concepts to applications. Addison-Wesley, Massachusetts.

CHANG, C.-T. (1994) Nonlinear dynamic Discontinuous Deformation Analysis with Finite Element meshed block system. *Ph.D. Thesis*, University of California, Berkeley, California.

CHERN, J.C., KOO, C.Y. and CHEN, S. (1995) Development of second order displacement function for DDA and manifold method. *Working Forum on the Manifold Method of Material Analysis*, Vicksburg. pp 183-202.

COOK, R.D., MALKUS, D.S. and PLESHA, M.E. (1989) Concepts and applications of Finite Element analysis. 3rd Edition. John Wiley and Sons, New York.

CUNDALL, P.A. (1971) A computer model for simulating progressive large-scale movements in blocky rock systems. *Symposium on Rock Fracture*, Nancy, France. Section 2 - 8.

DDA Version 96 (1996) © Gen-Hua Shi, 5650 Poinsett Ave., El Cerrito, CA.

de ARANTES, E.R. and OLIVEIRA, E. (1977) The patch test and the general convergence criteria of the Finite Element Method. *International Journal of Solids and Structures*, Vol 13, pp 159-178.

GOODMAN, R.E., TAYLOR, R. and BREKKE, T. (1968) A model for the mechanics of jointed rock. *Journal of the Soil Mechanics and Foundation Division of the American Society of Civil Engineers*, Vol 94, SM3, pp 637-659.

GOODMAN, R.E. and SHI, G.-H. (1985) Block theory and its applications to rock engineering. Prentice-Hall, Inc., New Jersey.

HAHN, J.K. (1988) Realistic animation of rigid bodies. *Computational Graphics*, Vol 22, pp 299-308.

HOEK, E. and BRAY, J.W. (1981) Rock slope engineering, 3rd Edition. The London Institute of Mining and Metallurgy.

KE, T.C. and GOODMAN, R.E. (1994) Discontinuous Deformation Analysis and the artificial joint concept. *Proceedings, 1st North American Rock Mechanics Symposium*, Austin. Balkema, Rotterdam. pp 607-614.

KE, T.C. (1996) The issue of rigid body rotation in DDA. *Proceedings, First International Forum on DDA and simulations of discontinuous media*, Berkeley, California. pp 318-325.

KOO, C.Y. and CHERN, J.C. (1996) The development of DDA with third order displacement function. *Proceedings, First International Forum on DDA and simulations of discontinuous media*, Berkeley, California. pp 342-349.

LIN, C.T. (1995) Extensions to the DDA method for jointed rock masses and other blocky systems. *Ph.D thesis*, University of Colorado, Boulder, Colorado.

MATHCAD PLUS 6.0. (1995) © Mathsoft Inc., 101 Main St., Cambridge, MA.

NDP C|C++ Compiler (1994) © Microway, Inc., Box 79, Kingston, MA.

SHI, G.-H. (1988) Discontinuous Deformation Analysis: A new numerical method for the statics and dynamics of block systems. *Ph.D. Thesis*, University of California, Berkeley, California.

WANG, F.D. and SUN, M.C. (1970) Slope stability analysis by Finite Element stress analysis and limiting equilibrium method. *U.S. Bureau of Mines Report of Investigations* 7341.

WILLIAMS, J.R., HOCKING, G. and MUSTOE, G.G.W. (1985) The theoretical basis of the discrete element method. In *NUMETA '85, Numerical Methods in Engineering, Theory and Applications*. Balkema, Rotterdam. pp 897-906.

YEUNG, M. (1991) Application of Shi's Discontinuous Deformation Analysis to rock slopes. *Ph.D thesis*, University of California, Berkeley, California.

YU, Y.S., GYENGE, M. and COATES, D.F. (1968) Comparison of stress and displacement in a gravity loaded slope by photoelasticity and Finite Element analysis. *Canadian Dept. Energy, Mines and Resources Report* MR 68-24 ID.

Appendix A

In this appendix, the analysis of a simple dynamic contact problem is set out. The aim of the analysis is to ensure that the transformation procedure set out in Chapter 5 is mathematically correct.

The geometry of the problem is shown in Figure A1. Block 1 is fixed at points A and B, and a point load acts at point C. Both bodies are subject to gravity loading in the negative y-direction, and block 2 has an initial velocity of –5 m/s in the y-direction. The details of loading and material properties are set out in the analysis files that follow. A single time increment of 0.01 seconds is considered.

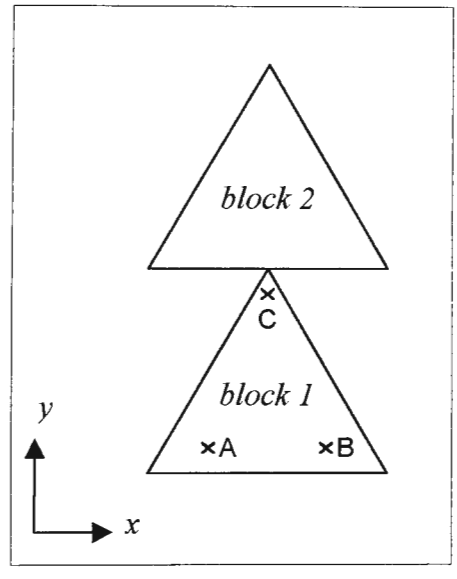


Figure A1: Schematic of the dynamic contact problem.

The two triangular blocks shown in Figure A1 can be modelled either as DDA blocks or as single triangular finite elements, and the results obtained should be the same in either case. To show that this is the case, a series of analyses are set out, and these are labelled A to D. Analyses A to C are sequential, being a single document written on Mathcad Plus 6.0 [1995], so that they draw on calculations from previous analyses.

- Analysis A (pages A2 to A10) is by the standard DDA method.

- Analysis B (pages A11 to A13) uses the solution equations from Analysis A as a starting point. These solution equations are now transformed so as to solve for the nodal displacements of block 1, and the deformation parameters of block 2.

- Analysis C (pages A14 to A15) again starts with the solution equations from Analysis A. In this case, however, the solution equations are transformed in order to solve for the nodal displacements of both blocks.

- Analysis D is performed using DDA Version 96 [1996]. The data file produced by program DDA Forward is reproduced (pages A16 to A17). This analysis is equivalent to Analysis A. Relevant results are found at the end of the data file, and are in bold text for emphasis.

It can be seen that the deformation parameters and block stresses obtained in all four analyses are identical.

ANALYSIS A

1. User Input

$E := 40 \cdot 10^9$ (Young's modulus)

$v := 0.3$ (Poisson's ratio)

$p := 1.6 \cdot 10^{12}$ (Penalty value)

$ne := 2$ (Number of elements)

$nn := 6$ (Number of nodes)

$nlp := 1$ (Number of load points)

$nfp := 2$ (Number of fixed points)

$\Delta t := 0.01$ (Time increment)

$$nset := \begin{bmatrix} 0 & 0 & 0 \\ 3.0 & 1.0 & 1 \\ 7.0 & 1.0 & 2 \\ 5.0 & 5.0 & 3 \\ 3.0 & 5.0 & 4 \\ 7.0 & 5.0 & 5 \\ 5.0 & 9.0 & 6 \end{bmatrix}$$

Node definition:
Column 1: x co-ordinate
Column 2: y co-ordinate
Column 3: node number

$$elset := \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 2 & 3 & 1 \\ 4 & 5 & 6 & 2 \end{pmatrix}$$

Element definition:
Column 1: 1st corner node
Column 2: 2nd corner node
Column 3: 3rd corner node
Column 4: element number

$$fp := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4.0 & 1.5 & 1 & 2 & 0.0 & 0.0 \\ 1 & 6.0 & 1.5 & 1 & 2 & 0.0 & 0.0 \end{pmatrix}$$

Fixed point definition:
Column 1: element number
Column 2: x co-ordinate
Column 3: y co-ordinate
Column 4: first d.o.f. in which the point is fixed
Column 5: last d.o.f. in which the point is fixed
Column 6: prescribed displacement in x direction
Column 7: prescribed displacement in y direction

$$lp := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 5.0 & 4.5 & 0 & -50 \cdot 10^6 \end{pmatrix}$$

Load point definition:
Column 1: element number
Column 2: x co-ordinate
Column 3: y co-ordinate
Column 4: load in x direction
Column 5: load in y direction

$$init := \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 3200 & 0 & -32000 & 0 & 0 \\ 3200 & 0 & -32000 & 0 & -5 \end{pmatrix}$$

Other properties:
(one row for each element)
Column 1: mass per unit area
Column 2: body force per unit area in x direction
Column 3: body force per unit area in y direction
Column 4: initial velocity in x direction
Column 5: initial velocity in y direction

---

2. Mesh Formulation

$i := 1 .. ne$

$x_{i,1} := nset_{(elset_{i,0}),0}$ $\quad$ $x_{i,6} := nset_{(elset_{i,0}),1}$

$x_{i,2} := nset_{(elset_{i,1}),0}$ $\quad$ $x_{i,7} := nset_{(elset_{i,1}),1}$

$x_{i,3} := nset_{(elset_{i,2}),0}$ $\quad$ $x_{i,8} := nset_{(elset_{i,2}),1}$

$x_{i,4} := nset_{(elset_{i,0}),0}$ $\quad$ $x_{i,9} := nset_{(elset_{i,0}),1}$

$i := 1 .. ne$

$$S_{i,0} := \frac{1}{2} \cdot \left[\sum_{k=1}^{3} \left(x_{i,k} \cdot x_{i,k+6} - x_{i,k+1} \cdot x_{i,k+5} \right) \right]$$

$$S_{i,1} := \frac{1}{6} \cdot \sum_{k=1}^{3} \left(x_{i,k} \cdot x_{i,k+6} - x_{i,k+1} \cdot x_{i,k+5} \right) \cdot \left(x_{i,k} + x_{i,k+1} \right)$$

$$S_{i,2} := \frac{1}{6} \cdot \sum_{k=1}^{3} \left(x_{i,k} \cdot x_{i,k+6} - x_{i,k+1} \cdot x_{i,k+5} \right) \cdot \left(x_{i,k+5} + x_{i,k+6} \right)$$

$$x_{i,0} := \frac{S_{i,1}}{S_{i,0}} \qquad x_{i,5} := \frac{S_{i,2}}{S_{i,0}}$$

$$x = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 5 & 3 & 7 & 5 & 3 & 2.33333 & 1 & 1 & 5 & 1 \\ 5 & 3 & 7 & 5 & 3 & 6.33333 & 5 & 5 & 9 & 5 \end{pmatrix}$$

$i := 1 .. ne$

$j := 1 .. 4$

$xb_{i,j} := x_{i,j} - x_{i,0}$

$xb_{i,j+5} := x_{i,j+5} - x_{i,5}$

$$xb = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -2 & 2 & 0 & -2 & 0 & -1.33333 & -1.33333 & 2.66667 & -1.33333 \\ 0 & -2 & 2 & 0 & -2 & 0 & -1.33333 & -1.33333 & 2.66667 & -1.33333 \end{pmatrix}$$

$$T(i, xp, yp) := \begin{bmatrix} 1 & 0 & -(yp - x_{i,5}) & (xp - x_{i,0}) & 0 & \frac{1}{2}\cdot(yp - x_{i,5}) \\ 0 & 1 & (xp - x_{i,0}) & 0 & (yp - x_{i,5}) & \frac{1}{2}\cdot(xp - x_{i,0}) \end{bmatrix}$$

3. Simplexes

$i := 1 .. ne$

$$S_{i,0} := \frac{1}{2}\cdot\left[\sum_{k=1}^{3}\left(xb_{i,k}\cdot xb_{i,k+6} - xb_{i,k+1}\cdot xb_{i,k+5}\right)\right]$$

$$S_{i,1} := \frac{1}{6}\cdot\sum_{k=1}^{3}\left(xb_{i,k}\cdot xb_{i,k+6} - xb_{i,k+1}\cdot xb_{i,k+5}\right)\cdot\left(xb_{i,k} + xb_{i,k+1}\right)$$

$$S_{i,2} := \frac{1}{6}\cdot\sum_{k=1}^{3}\left(xb_{i,k}\cdot xb_{i,k+6} - xb_{i,k+1}\cdot xb_{i,k+5}\right)\cdot\left(xb_{i,k+5} + xb_{i,k+6}\right)$$

$$S_{i,3} := \frac{1}{12}\cdot\sum_{k=1}^{3}\left(xb_{i,k}\cdot xb_{i,k+6} - xb_{i,k+1}\cdot xb_{i,k+5}\right)\cdot\left[\left(xb_{i,k}\right)^2 + xb_{i,k}\cdot xb_{i,k+1} + \left(xb_{i,k+1}\right)^2\right]$$

$$S_{i,4} := \frac{1}{12} \cdot \sum_{k=1}^{3} \left(xb_{i,k} \cdot xb_{i,k+6} - xb_{i,k+1} \cdot xb_{i,k+5}\right) \cdot \left[\left(xb_{i,k+5}\right)^2 + xb_{i,k+5} \cdot xb_{i,k+6} + \left(xb_{i,k+6}\right)^2\right]$$

$$S_{i,5} := \frac{1}{24} \cdot \sum_{k=1}^{3} \left(xb_{i,k} \cdot xb_{i,k+6} - xb_{i,k+1} \cdot xb_{i,k+5}\right) \cdot \left(2 \cdot xb_{i,k} \cdot xb_{i,k+5} + xb_{i,k} \cdot xb_{i,k+6} + xb_{i,k+1} \cdot xb_{i,k+5} +\right.$$

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 8 & 0 & 2.36848 \cdot 10^{-15} & 5.33333 & 7.11111 & 0 \\ 8 & 0 & 2.36848 \cdot 10^{-15} & 5.33333 & 7.11111 & 0 \end{bmatrix}$$

---

4. DDA Solution Equation Formulation

4.1 Initialise stiffness matrix and load vector

$m := ne \cdot 6 - 1$

$i := 0..m$

$j := 0..m$

$K_{i,j} := 0$

$F_j := 0$

4.2 Inertia term

$$Km(i) := \begin{bmatrix} S_{i,0} & 0 & 0 & 0 & 0 & 0 \\ 0 & S_{i,0} & 0 & 0 & 0 & 0 \\ 0 & 0 & S_{i,3} + S_{i,4} & -S_{i,5} & S_{i,5} & \frac{S_{i,3} - S_{i,4}}{2} \\ 0 & 0 & -S_{i,5} & S_{i,3} & 0 & \frac{S_{i,5}}{2} \\ 0 & 0 & S_{i,5} & 0 & S_{i,4} & \frac{S_{i,5}}{2} \\ 0 & 0 & \frac{S_{i,3} - S_{i,4}}{2} & \frac{S_{i,5}}{2} & \frac{S_{i,5}}{2} & \frac{S_{i,3} + S_{i,4}}{4} \end{bmatrix}$$

$i := 1 .. ne$

$j := 0 .. 5$

$V_{i,j} := 0$

$i := 1 .. ne$

$V_{i,0} := init_{i,3}$

$V_{i,1} := init_{i,4}$

$j := 0 .. 5$

$k := 0 .. 5$

$$K_{(i-1)\cdot 6+j,(i-1)\cdot 6+k} := K_{(i-1)\cdot 6+j,(i-1)\cdot 6+k} + 2\cdot\frac{init_{i,0}}{\Delta t^2}\cdot Km(i)_{j,k}$$

$$F_{(i-1)\cdot 6+j} := F_{(i-1)\cdot 6+j} + 2\cdot\frac{init_{i,0}}{\Delta t}\cdot Km(i)_{j,k}\cdot V_{i,k}$$

4.3 Prescribed displacement

$n := 1 .. nfp$

$i := 0 .. 5$

$j := 0 .. 5$

$$K_{(fp_{n,0}-1)\cdot 6+i,(fp_{n,0}-1)\cdot 6+j} := K_{(fp_{n,0}-1)\cdot 6+i,(fp_{n,0}-1)\cdot 6+j} + 100\cdot p\cdot \sum_{k=fp_{n,3}-1}^{fp_{n,4}-1} T\left(fp_{n,}\right.$$

$$F_{(fp_{n,0}-1)\cdot 6+i} := F_{(fp_{n,0}-1)\cdot 6+i} + 100\cdot p\cdot \sum_{k=fp_{n,3}-1}^{fp_{n,4}-1} T\left(fp_{n,0}, fp_{n,1}, fp_{n,2}\right)_{k,i}\cdot fp_{n,k+5}$$

4.4 Elastic strain matrix

$$Ke(i) := \frac{E}{1-\nu^2} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \nu & 0 \\ 0 & 0 & 0 & \nu & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \cdot S_{i,0}$$

$i := 1 .. ne$

$j := 0 .. 5$

$k := 0 .. 5$

$$K_{(i-1)\cdot 6+j,(i-1)\cdot 6+k} := K_{(i-1)\cdot 6+j,(i-1)\cdot 6+k} + Ke(i)_{j,k}$$

4.5 Point load

$n := 1 .. nlp$

$i := 0 .. 5$

$j := 0 .. 1$

$$F_{(lp_{n,0}-1)\cdot 6+i} := F_{(lp_{n,0}-1)\cdot 6+i} + T\left(lp_{n,0}, lp_{n,1}, lp_{n,2}\right)_{j,i} \cdot lp_{n,j+3}$$

4.6 Volume Loading

$j := 1 .. ne$

$$F_{(j-1)\cdot 6} := F_{(j-1)\cdot 6} + S_{j,0} \cdot init_{j,1}$$

$$F_{(j-1)\cdot 6+1} := F_{(j-1)\cdot 6+1} + S_{j,0} \cdot init_{j,2}$$

4.7. Contact

$x_1 := 5 \qquad x_2 := 7 \qquad x_3 := 3$

$y_1 := 5 \qquad y_2 := 5 \qquad y_3 := 5$

$$S0 := \left\| \begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix} \right\|$$

$$l := \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$$

$$e := \frac{1}{l} \cdot T(1, x_1, y_1)^T \cdot \begin{pmatrix} y_2 - y_3 \\ x_3 - x_2 \end{pmatrix}$$

$$g := \frac{1}{l} \cdot T(2, x_2, y_2)^T \cdot \begin{pmatrix} y_3 - y_1 \\ x_1 - x_3 \end{pmatrix} + \frac{1}{l} \cdot T(2, x_3, y_3)^T \cdot \begin{pmatrix} y_1 - y_2 \\ x_2 - x_1 \end{pmatrix}$$

$i := 0..5$

$j := 0..5$

$K_{i,j} := K_{i,j} + p \cdot e_i \cdot e_j$

$K_{i+6,j+6} := K_{i+6,j+6} + p \cdot g_i \cdot g_j$

$K_{i,j+6} := K_{i,j+6} + p \cdot e_i \cdot g_j$

$K_{i+6,j} := K_{i+6,j} + p \cdot g_i \cdot e_j$

$F_i := F_i - p \cdot \frac{S0}{l} \cdot e_i$

$F_{i+6} := F_{i+6} - p \cdot \frac{S0}{l} \cdot g_i$

5. Solve Solution Equations

$$d := K^{-1} \cdot F$$

6. Derive Element Deformations, Nodal Displacements and Stresses

$$dl_i := d_i$$

$$d2_i := d_{i+6}$$

$i := 0..2$

$$ul_{i\cdot 2} := \sum_{j=0}^{5} T\left[1, nset_{\left(elset_{1,i}\right),0}, nset_{\left(elset_{1,i}\right),1}\right]_{0,j} \cdot dl_j$$

$$ul_{i\cdot 2+1} := \sum_{j=0}^{5} T\left[1, nset_{\left(elset_{1,i}\right),0}, nset_{\left(elset_{1,i}\right),1}\right]_{1,j} \cdot dl_j$$

$$u2_{i\cdot 2} := \sum_{j=0}^{5} T\left[2, nset_{\left(elset_{2,i}\right),0}, nset_{\left(elset_{2,i}\right),1}\right]_{0,j} \cdot d2_j$$

$$u2_{i\cdot 2+1} := \sum_{j=0}^{5} T\left[2, nset_{\left(elset_{2,i}\right),0}, nset_{\left(elset_{2,i}\right),1}\right]_{1,j} \cdot d2_j$$

$i := 0..5$

$$\sigma 1_i := 0$$

$$\sigma 2_i := 0$$

$j := 0..5$

$$\sigma 1_i := \frac{Ke(1)_{i,j}}{S_{1,0}} \cdot dl_j + \sigma 1_i$$

$$\sigma 2_i := \frac{Ke(2)_{i,j}}{S_{2,0}} \cdot d2_j + \sigma 2_i$$

7. Results:

$$d1 = \begin{bmatrix} 0 \\ -5.58823 \cdot 10^{-4} \\ 0 \\ 2.20738 \cdot 10^{-7} \\ -6.70308 \cdot 10^{-4} \\ 0 \end{bmatrix} \quad u1 = \begin{bmatrix} -4.41476 \cdot 10^{-7} \\ 3.34921 \cdot 10^{-4} \\ 4.41476 \cdot 10^{-7} \\ 3.34921 \cdot 10^{-4} \\ 0 \\ -2.34631 \cdot 10^{-3} \end{bmatrix} \quad \sigma 1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -8.82952 \cdot 10^{6} \\ -2.94612 \cdot 10^{7} \\ 0 \end{bmatrix}$$

$$d2 = \begin{bmatrix} 0 \\ -2.498 \cdot 10^{-3} \\ 0 \\ 3.0645 \cdot 10^{-5} \\ -1.02249 \cdot 10^{-4} \\ 0 \end{bmatrix} \quad u2 = \begin{bmatrix} -6.12899 \cdot 10^{-5} \\ -2.36167 \cdot 10^{-3} \\ 6.12899 \cdot 10^{-5} \\ -2.36167 \cdot 10^{-3} \\ 0 \\ -2.77067 \cdot 10^{-3} \end{bmatrix} \quad \sigma 2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.30752 \cdot 10^{3} \\ -4.09035 \cdot 10^{6} \\ 0 \end{bmatrix}$$

==========

ANALYSIS B

5b. Derive Q- Matrices

$Q := \text{for } m \in 1..ne$

$\quad$ for $i \in 0..1$

$\quad\quad$ for $j \in 0..5$

$\quad\quad\quad R_{i,j} \leftarrow T\left[m, nset_{\left(elset_{m,0}\right),0}, nset_{\left(elset_{m,0}\right),1}\right]_{i,j}$

$\quad\quad\quad R_{i+2,j} \leftarrow T\left[m, nset_{\left(elset_{m,1}\right),0}, nset_{\left(elset_{m,1}\right),1}\right]_{i,j}$

$\quad\quad\quad R_{i+4,j} \leftarrow T\left[m, nset_{\left(elset_{m,2}\right),0}, nset_{\left(elset_{m,2}\right),1}\right]_{i,j}$

$\quad Qm \leftarrow R^{-1}$

$\quad$ for $i \in 0..5$

$\quad\quad$ for $j \in 0..5$

$\quad\quad\quad Q_{i,(m-1)\cdot 6+j} \leftarrow Qm_{i,j}$

$\quad Q$

6b. Transform Solution Equations

(for convenience the stiffness matrix and load vectors are first split up into their individual elements)

$i := 0..11$

$j := 0..11$

$KT_{i,j} := 0$

$FT_i := 0$

$i := 0..5$

$j := 0..5$

$K11_{i,j} := K_{i,j}$

$K12_{i,j} := K_{i,j+6}$

$K21_{i,j} := K_{i+6,j}$

$K22_{i,j} := K_{i+6,j+6}$

$F1_i := F_i$

$F2_i := F_{i+6}$

$Q1_{i,j} := Q_{i,j}$

$Q2_{i,j} := Q_{i,j+6}$

$KT11 := Q1^T \cdot K11 \cdot Q1$

$KT12 := Q1^T \cdot K12$

$KT21 := K21 \cdot Q1$

$KT22 := K22$

$FT1 := Q1^T \cdot F1$

$FT2 := F2$

$i := 0..5$

$j := 0..5$

$KT_{i,j} := KT11_{i,j}$

$KT_{i,j+6} := KT12_{i,j}$

$KT_{i+6,j} := KT21_{i,j}$

$KT_{i+6,j+6} := KT22_{i,j}$

$FT_i := FT1_i$

$FT_{i+6} := FT2_i$

7b. Solve Solution Equations

$u := KT^{-1} \cdot FT$

8b. Derive element deformation variables, nodal displacements and stresses

$i := 0..5$

$ul_i := u_i$

$d2_i := u_{i+6}$

$d1 := Q1 \cdot ul$

$u2 := Q2^{-1} \cdot d2$

$\sigma 1 := \frac{1}{S_{1,0}} \cdot Ke(1) \cdot d1$

$\sigma 2 := \frac{1}{S_{2,0}} \cdot Ke(2) \cdot d2$

9b. Results

$$d1 = \begin{bmatrix} 0 \\ -5.58823 \cdot 10^{-4} \\ 0 \\ 2.20738 \cdot 10^{-7} \\ -6.70308 \cdot 10^{-4} \\ 0 \end{bmatrix} \quad ul = \begin{bmatrix} -4.41476 \cdot 10^{-7} \\ 3.34921 \cdot 10^{-4} \\ 4.41476 \cdot 10^{-7} \\ 3.34921 \cdot 10^{-4} \\ 0 \\ -2.34631 \cdot 10^{-3} \end{bmatrix} \quad \sigma 1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -8.82952 \cdot 10^{6} \\ -2.94612 \cdot 10^{7} \\ 1.34177 \cdot 10^{-9} \end{bmatrix}$$

$$d2 = \begin{bmatrix} 0 \\ -2.498 \cdot 10^{-3} \\ 0 \\ 3.0645 \cdot 10^{-5} \\ -1.02249 \cdot 10^{-4} \\ 0 \end{bmatrix} \quad u2 = \begin{bmatrix} -6.12899 \cdot 10^{-5} \\ -2.36167 \cdot 10^{-3} \\ 6.12899 \cdot 10^{-5} \\ -2.36167 \cdot 10^{-3} \\ 0 \\ -2.77067 \cdot 10^{-3} \end{bmatrix} \quad \sigma 2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.30752 \cdot 10^{3} \\ -4.09035 \cdot 10^{6} \\ 0 \end{bmatrix}$$

==========

ANALYSIS C

6c. Convert Solution Equations

$$KT11 := Q1^{T} \cdot K11 \cdot Q1$$

$$KT12 := Q1^{T} \cdot K12 \cdot Q2$$

$$KT21 := Q2^{T} \cdot K21 \cdot Q1$$

$$KT22 := Q2^{T} \cdot K22 \cdot Q2$$

$$FT1 := Q1^{T} \cdot F1$$

$$FT2 := Q2^{T} \cdot F2$$

$$i := 0..5$$

$$j := 0..5$$

$$KT_{i,j} := KT11_{i,j}$$

$$KT_{i,j+6} := KT12_{i,j}$$

$$KT_{i+6,j} := KT21_{i,j}$$

$$KT_{i+6,j+6} := KT22_{i,j}$$

$$FT_{i} := FT1_{i}$$

$$FT_{i+6} := FT2_{i}$$

7c. Solve Solution Equations

$$u := KT^{-1} \cdot FT$$

8c. Derive element deformation variables, nodal displacements and stresses

$$i := 0..5$$

$$u1_{i} := u_{i}$$

$$u2_{i} := u_{i+6}$$

$$d1 := Q1 \cdot u1$$

$$d2 := Q2 \cdot u2$$

$$\sigma 1 := \frac{1}{S_{1,0}} \cdot Ke(1) \cdot d1$$

$$\sigma 2 := \frac{1}{S_{2,0}} \cdot Ke(2) \cdot d2$$

9c. Results

$$d1 = \begin{bmatrix} 0 \\ -5.58823 \cdot 10^{-4} \\ 0 \\ 2.20738 \cdot 10^{-7} \\ -6.70308 \cdot 10^{-4} \\ 0 \end{bmatrix} \qquad u1 = \begin{bmatrix} -4.41476 \cdot 10^{-7} \\ 3.34921 \cdot 10^{-4} \\ 4.41476 \cdot 10^{-7} \\ 3.34921 \cdot 10^{-4} \\ 0 \\ -2.34631 \cdot 10^{-3} \end{bmatrix} \qquad \sigma 1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -8.82952 \cdot 10^{6} \\ -2.94612 \cdot 10^{7} \\ -4.89145 \cdot 10^{-9} \end{bmatrix}$$

$$d2 = \begin{bmatrix} 0 \\ -2.498 \cdot 10^{-3} \\ 0 \\ 3.0645 \cdot 10^{-5} \\ -1.02249 \cdot 10^{-4} \\ 0 \end{bmatrix} \qquad u2 = \begin{bmatrix} -6.12899 \cdot 10^{-5} \\ -2.36167 \cdot 10^{-3} \\ 6.12899 \cdot 10^{-5} \\ -2.36167 \cdot 10^{-3} \\ 0 \\ -2.77067 \cdot 10^{-3} \end{bmatrix} \qquad \sigma 2 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1.30752 \cdot 10^{3} \\ -4.09035 \cdot 10^{6} \\ -2.90289 \cdot 10^{-8} \end{bmatrix}$$

==========

ANALYSIS D

```
###### average  block  area ###### 8.000000
###### average  block  area ###### 8.000000
###### minimum  edge length ###### 4.000000
###### minimum  edge length ###### 4.000000
###### minimum v-e distance ###### 3.577709
###### minimum v-e distance ###### 3.577709
###### minimum block  angle ###### 53.130105
###### minimum block  angle ###### 53.130105
enter 0 or 1, 0-statics 1-dynamics
enter number of time steps (1-100)
enter number of block materials
enter number of joint materials
enter max. allowable step displacement divided
by half height of whole block mesh (.02-.0001)
enter upper limit of time interval per step
enter 0 choose automatic time step chosen
enter stiffness of contact spring
enter 0 choose automatic spring stiffness
enter time step number >=2 for each fixed load
point i to have time dependent (t u v) t=time
enter 0 to set time step 2 u=0 v=0 all time
enter time depending movement & loads
enter block material constants
enter joint material constants
enter factor of SOR (1 -- 2) ?
number of blocks: 2
number of steps : 1
fixed points    : 2
loading points  : 1
measured points : 2
block materials : 2
joint materials : 1
time interval   : 0.010000
step disp. ratio: 0.010000
###### aver blk area /w0/w0 ###### 0.453515
###### min edge length  /w0 ###### 0.952381
###### min v-e distance /w0 ###### 0.851835
###### minimum block  angle ###### 53.130105
### d0-in: 0.105000 d0-ge 0.533333 ###
### d0-in: 0.105000 d0-ge 0.533333 ###
limx=1023 limy= 767 max_color=255
m point 1 block 1 u -0.000000 v -0.000000 ex 0.000000 ey
0.000000 exy 0.000000
m point 1 block 1 cx 0.000000 cy 0.000000 cxy 0.000000
m point 2 block 2 u -0.000000 v -0.000000 ex 0.000000 ey
0.000000 exy 0.000000
m point 2 block 2 cx 0.000000 cy 0.000000 cxy 0.000000
<<< step >>>     1
iterate   1      1
close point      0
open  point      0
rel max ds 0.06597
step  time 0.01000
total time 0.01000
sprin stif 1600000000000.000000
```

```
<<< step >>>      1
iterate   1       1
close point       0
open  point       0
rel max ds 0.06597
step time  0.01000
!!!!!step!!!!!    1
!!!!!step!!!!!    1
contact=       1 transfer=       0 close=       1
<<< step >>>      1
iterate  -1       2
close point       0
open  point       0
rel max ds 0.06597
step  time 0.01000
total time 0.01000
sprin stif 1600000000000.000000
<<< step >>>      1
iterate  -1       2
close point       0
open  point       0
rel max ds 0.06597
step time  0.01000
wf/d0 -0.000146 i 1
block deformation and stress
u0      v1       az
ex      ey      gxy
cx      cy      txy

m point 1 block 1 u 0.000000 v -0.000559 ex 0.000000 ey -
0.000670 exy 0.000000
m point 1 block 1 cx -8829523.551889 cy -29461176.886742
cxy 0.053213
m point 2 block 2 u -0.000000 v -0.002498 ex 0.000031 ey -
0.000102 exy 0.000000
m point 2 block 2 cx -1307.518441 cy -4090353.523569 cxy
0.008731
itereration       2
```

Appendix B

In this appendix, a Mathcad Plus 6.0 [1995] file is reproduced. The file is designed to perform static, single time-step analyses on single blocks sub-meshed using 1st order elements. The sub-meshing method is that described in Chapter 5, and interaction with other blocks is not considered.

The example file shown is an analysis of the Cook cantilever test (Section 5.8.3), where the cantilever is composed of 8 1st order elements. The geometry and meshing of the problem is shown in Figure B1.

The file is a general file, however, and other problems can be solved by altering the input in Section 1.

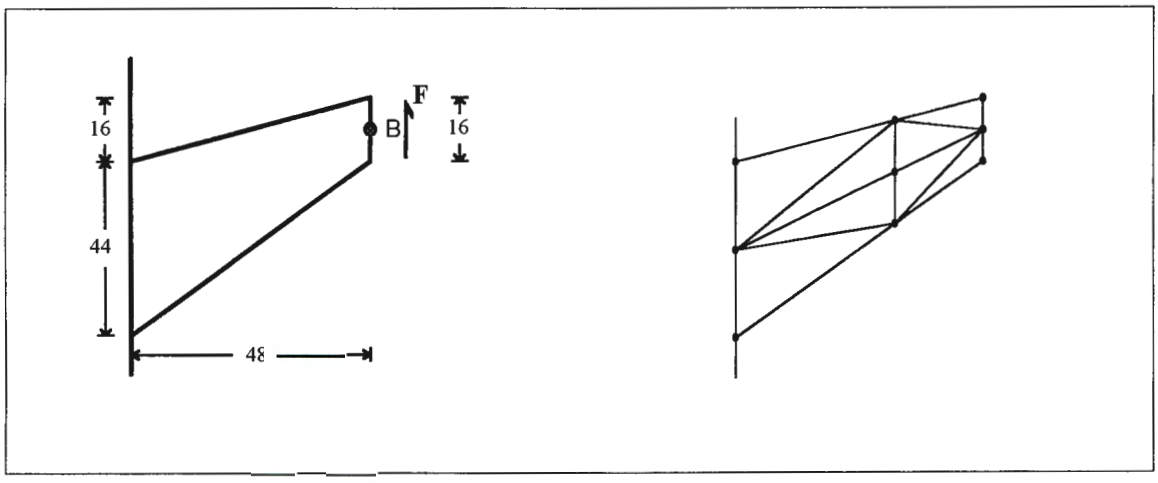


Figure B1: Geometry and meshing of the Cook cantilever test.

MESHING: COOK CANTILEVER TEST USING 8 1st ORDER ELEMENTS

1. User Input

$E_y := 40000$ (Young's modulus)

$v := 0.3$ (Poisson's ratio)

$p := 900000$ (Penalty value)

$ne := 8$ (Number of elements)

$nn := 9$ (Number of nodes)

$nlp := 3$ (Number of load points)

$nfp := 3$ (Number of fixed points)

$$nset := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 44 & 1 \\ 35.2 & 55.733 & 2 \\ 48 & 60 & 3 \\ 0 & 22 & 4 \\ 35.2 & 44 & 5 \\ 48 & 52 & 6 \\ 0 & 0 & 7 \\ 35.2 & 32.267 & 8 \\ 48 & 44 & 9 \end{bmatrix} \quad elset := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 4 & 2 & 1 \\ 2 & 4 & 5 & 2 \\ 2 & 5 & 6 & 3 \\ 2 & 6 & 3 & 4 \\ 4 & 7 & 8 & 5 \\ 4 & 8 & 5 & 6 \\ 5 & 8 & 6 & 7 \\ 6 & 8 & 9 & 8 \end{bmatrix}$$

Node definition (nset):
Column 1: x co-ordinate
Column 2: y co-ordinate
Column 3: node number

Element definition (elset):
Column 1: node 1
Column 2: node 2
Column 3: node 3
Column 4: element number

$$fp := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 \\ 1 & 2 & 1 & 2 & 0 & 0 \\ 5 & 2 & 1 & 2 & 0 & 0 \end{bmatrix}$$

Fixed point definition:
Column 1: element number
Column 2: node number
Column 3: first d.o.f. in which point is fixed
Column 4: last d.o.f. in which point is fixed
Column 5: prescribed displacement, u_m
Column 6: prescribed displacement v_m

$$lp := \begin{bmatrix} 0 & 0 & 0 & 0 \\ 8 & 3 & 0 & 25 \\ 7 & 3 & 0 & 50 \\ 4 & 3 & 0 & 25 \end{bmatrix}$$

Load point definition:
Column 1: element number
Column 2: node number
Column 3: load in x direction
Column 4: load in y direction

2. Mesh Formulation

$i := 1 .. ne$

$x_{i,1} := nset_{(elset_{i,0}),0}$ $\quad$ $x_{i,6} := nset_{(elset_{i,0}),1}$

$x_{i,2} := nset_{(elset_{i,1}),0}$ $\quad$ $x_{i,7} := nset_{(elset_{i,1}),1}$

$x_{i,3} := nset_{(elset_{i,2}),0}$ $\quad$ $x_{i,8} := nset_{(elset_{i,2}),1}$

$x_{i,4} := x_{i,1}$ $\quad$ $x_{i,9} := x_{i,6}$

$x_{i,0} := \frac{x_{i,1} + x_{i,2} + x_{i,3}}{3}$ $\quad$ $x_{i,5} := \frac{x_{i,6} + x_{i,7} + x_{i,8}}{3}$

$x =$

	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	11.733	0	0	35.2	0	40.578	44	22
2	23.467	35.2	0	35.2	35.2	40.578	55.733	22
3	39.467	35.2	35.2	48	35.2	50.578	55.733	44
4	43.733	35.2	48	48	35.2	55.911	55.733	52
5	11.733	0	0	35.2	0	18.089	22	0
6	23.467	0	35.2	35.2	0	32.756	22	32.267
7	39.467	35.2	35.2	48	35.2	42.756	44	32.267
8	43.733	48	35.2	48	48	42.756	52	32.267

$$T(i,j) := \begin{bmatrix} 1 & 0 & -\left(x_{i,j+5} - x_{i,5}\right) & \left(x_{i,j} - x_{i,0}\right) & 0 & \frac{1}{2}\cdot\left(x_{i,j+5} - x_{i,5}\right) \\ 0 & 1 & \left(x_{i,j} - x_{i,0}\right) & 0 & \left(x_{i,j+5} - x_{i,5}\right) & \frac{1}{2}\cdot\left(x_{i,j} - x_{i,0}\right) \end{bmatrix}$$

3. Simplexes

$i := 1 .. ne$

$$S_i := \frac{1}{2} \cdot \left[\sum_{k=1}^{3} \left(x_{i,k} \cdot x_{i,k+6} - x_{i,k+1} \cdot x_{i,k+5} \right) \right]$$

$$\sum_{i=1}^{8} S_i = 1.44 \cdot 10^3$$

4. DDA Stiffness Matrix Formulation

4.1 Initialise

$m := ne \cdot 6 - 1$

$i := 1 .. m$

$j := 1 .. ne$

$k := 0 .. 5$

$K_{k,i} := 0$

$F_{k,j} := 0$

4.2 Elastic strain matrix

$$E := \frac{E_y}{1 - v^2} \cdot \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & v & 0 \\ 0 & 0 & 0 & v & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1 - v}{2} \end{bmatrix}$$

$i := 1 .. ne$

$j := 0 .. 5$

$k := 0 .. 5$

$$K_{k,(i-1)\cdot 6+j} := K_{k,(i-1)\cdot 6+j} + S_i \cdot E_{k,j}$$

4.3 Point load

$n := 1 .. nlp$

$i := 0 .. 5$

$j := 0 .. 1$

$$F_{i,(lp_{n,0})} := F_{i,(lp_{n,0})} + T\left(lp_{n,0}, lp_{n,1}\right)_{j,i} \cdot lp_{n,j+2}$$

4.4 Prescribed displacement

$n := 1 .. nfp$

$i := 0 .. 5$

$j := 0 .. 5$

$$K_{i,(fp_{n,0}-1)\cdot 6+j} := K_{i,(fp_{n,0}-1)\cdot 6+j} + 100 \cdot p \cdot \sum_{k=fp_{n,2}-1}^{fp_{n,3}-1} T\left(fp_{n,0}, fp_{n,1}\right)_{k,i} \cdot T\left(fp_{n,0}, fp_{n,1}\right)_{k,j}$$

$$F_{i,fp_{n,0}} := F_{i,fp_{n,0}} + 100 \cdot p \cdot \sum_{k=fp_{n,2}-1}^{fp_{n,3}-1} T\left(fp_{n,0}, fp_{n,1}\right)_{k,i} \cdot fp_{n,k+4}$$

$K =$

	41	42	43	44	45	46
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	$2.251 \cdot 10^6$	$6.752 \cdot 10^5$
4	0	0	0	0	$6.752 \cdot 10^5$	$2.251 \cdot 10^6$
5	$1.155 \cdot 10^6$	0	0	0	0	0

$$F = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25 & 0 & 0 & 50 & 25 \\ 0 & 0 & 0 & 0 & 106.667 & 0 & 0 & 426.667 & 106.667 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 102.225 & 0 & 0 & 462.217 & 31.108 \\ 0 & 0 & 0 & 0 & 53.333 & 0 & 0 & 213.333 & 53.333 \end{bmatrix}$$

5. Derive Q- Matrices

$Q := \text{for } m \in 1..ne$

  for $i \in 0..1$

    for $j \in 0..5$

      $R_{i,j} \leftarrow T(m,1)_{i,j}$

      $R_{i+2,j} \leftarrow T(m,2)_{i,j}$

      $R_{i+4,j} \leftarrow T(m,3)_{i,j}$

  $Qm \leftarrow R^{-1}$

  for $i \in 0..5$

    for $j \in 0..5$

      $Q_{i,(m-1)\cdot 6+j} \leftarrow Qm_{i,j}$

  Q

$Q =$

	0	1	2	3	4	5	6
0	0.333	0	0.333	0	0.333	0	0.333
1	0	0.333	0	0.333	0	0.333	0
2	-0.023	-0.022	0.023	$7.576 \cdot 10^{-3}$	0	0.014	-0.043
3	-0.044	0	0.015	0	0.028	0	-0.053
4	0	0.045	0	-0.045	0	0	0
5	0.045	-0.044	-0.045	0.015	0	0.028	0.085

6. Transform K- and F-matrices into FE format

$m := 1..8$

$i := 0..5$

$j := 0..5$

$$KT_{i,(m-1)\cdot 6+j} := \sum_{k=0}^{5}\left[\sum_{l=0}^{5}\left[Q_{k,(m-1)\cdot 6+i}\cdot K_{k,(m-1)\cdot 6+l}\cdot Q_{l,(m-1)\cdot 6+j}\right]\right]$$

$$FT_{i,m} := \sum_{j=0}^{5} Q_{j,(m-1)\cdot 6+i}\cdot F_{j,m}$$

$KT =$

	43	44	45	46
0	$-2.095\cdot 10^{4}$	$2.015\cdot 10^{4}$	$-7.692\cdot 10^{3}$	$-6.2\cdot 10^{4}$
1	$4.551\cdot 10^{4}$	$-6.593\cdot 10^{3}$	$7.051\cdot 10^{3}$	$2.755\cdot 10^{4}$
2	$-6.593\cdot 10^{3}$	$1.374\cdot 10^{4}$	$1.525\cdot 10^{-12}$	$-3.388\cdot 10^{4}$
3	$7.051\cdot 10^{3}$	$1.525\cdot 10^{-12}$	$4.808\cdot 10^{3}$	$7.692\cdot 10^{3}$
4	$2.755\cdot 10^{4}$	$-3.388\cdot 10^{4}$	$7.692\cdot 10^{3}$	$9.588\cdot 10^{4}$
5	$-5.256\cdot 10^{4}$	$6.593\cdot 10^{3}$	$-1.186\cdot 10^{4}$	$-3.524\cdot 10^{4}$

$$FT = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.776\cdot 10^{-15} & 0 & 0 & -1.066\cdot 10^{-14} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.332\cdot 10^{-15} & 0 & 0 & -1.776\cdot 10^{-15} & -1.776\cdot 10^{-15} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 25 & 0 & 0 & 50 & 25 \end{bmatrix}$$

7. Initialise and Assemble Global Stiffness Matrix

$i := 0..17$

$j := 0..17$

$KF_{i,j} := 0 \qquad FF_i := 0$

$m := 1..8$

$i := 0..1$

$j := 0..1$

$$KF\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right] := KF\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j + KT_{i,(m-1)\cdot 6+j}$$

$$KF\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right] := KF\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j + KT_{i,(m-1)\cdot 6+2+j}$$

$$KF\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right] := KF\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j + KT_{i,(m-1)\cdot 6+4+j}$$

$$KF\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right] := KF\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j + KT_{i+2,(m-1)\cdot 6+j}$$

$$KF\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right] := KF\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j + KT_{i+2,(m-1)\cdot 6+2+j}$$

$$KF\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right] := KF\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j + KT_{i+2,(m-1)\cdot 6+4+j}$$

$$KF\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right] := KF\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j + KT_{i+4,(m-1)\cdot 6+j}$$

$$KF\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right] := KF\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j + KT_{i+4,(m-1)\cdot 6+2+j}$$

$$KF\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right] := KF\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j + KT_{i+4,(m-1)\cdot 6+4+j}$$

$$FF_{\left(elset_{m,0}-1\right)\cdot 2+i} := FF_{\left(elset_{m,0}-1\right)\cdot 2+i} + FT_{i,m}$$

$$FF_{\left(elset_{m,1}-1\right)\cdot 2+i} := FF_{\left(elset_{m,1}-1\right)\cdot 2+i} + FT_{i+2,m}$$

$$FF_{\left(elset_{m,2}-1\right)\cdot 2+i} := FF_{\left(elset_{m,2}-1\right)\cdot 2+i} + FT_{i+4,m}$$

KF =

	14	15	16	17
8	$-6.659\cdot10^4$	$3.653\cdot10^4$	0	0
9	$3.653\cdot10^4$	$-1.022\cdot10^5$	0	0
10	$3.388\cdot10^4$	$-1.429\cdot10^4$	$-6.2\cdot10^4$	$2.864\cdot10^4$
11	$-1.429\cdot10^4$	$1.186\cdot10^4$	$2.755\cdot10^4$	$-5.256\cdot10^4$
12	$6.41\cdot10^3$	$-7.692\cdot10^3$	0	0
13	$-6.593\cdot10^3$	$2.244\cdot10^3$	0	0
14	$9.406\cdot10^4$	$-3.653\cdot10^4$	$-3.388\cdot10^4$	$6.593\cdot10^3$
15	$-3.653\cdot10^4$	$1.118\cdot10^5$	$7.692\cdot10^3$	$-1.186\cdot10^4$
16	$-3.388\cdot10^4$	$7.692\cdot10^3$	$9.588\cdot10^4$	$-3.524\cdot10^4$
17	$6.593\cdot10^3$	$-1.186\cdot10^4$	$-3.524\cdot10^4$	$6.442\cdot10^4$

FF =

	0
6	0
7	0
8	0
9	$-1.066\cdot10^{-14}$
10	0
11	50
12	0
13	0
14	0
15	$-3.553\cdot10^{-15}$
16	0
17	25

8. Solve for Nodal Displacements

$KFI := KF^{-1}$

$ndof := nn\cdot 2 - 1$

$i := 0..ndof$

$$U_{i,1} := \sum_{j=0}^{ndof} KFI_{i,j}\cdot FF_j \qquad U_{i,0} := floor\left(\frac{i}{2}+1\right)$$

$$U = \begin{bmatrix} 1 & -1.705\cdot10^{-6} \\ 1 & 1.048\cdot10^{-6} \\ 2 & -3.879\cdot10^{-3} \\ 2 & 9.339\cdot10^{-3} \\ 3 & -8.037\cdot10^{-3} \\ 3 & 0.018 \\ 4 & 9.852\cdot10^{-7} \\ 4 & 2.29\cdot10^{-7} \\ 5 & -1.668\cdot10^{-3} \\ 5 & 9.49\cdot10^{-3} \\ 6 & -3.236\cdot10^{-3} \\ 6 & 0.018 \\ 7 & 7.195\cdot10^{-7} \\ 7 & -1.66\cdot10^{-7} \\ 8 & 8.346\cdot10^{-4} \\ 8 & 9.113\cdot10^{-3} \\ 9 & -1.316\cdot10^{-3} \\ 9 & 0.017 \end{bmatrix}$$

9. Transform Back to DDA Block Unknowns, & Compute Stresses.

$i := 1 .. ne$

$j := 0 .. 5$

$$d_{i-1,j+1} := \sum_{k=0}^{2} Q_{j,(i-1)\cdot 6+2\cdot k} \cdot U_{(elset_{i,k}-1)\cdot 2,1} + \sum_{k=0}^{2} Q_{j,(i-1)\cdot 6+2\cdot k+1} \cdot U_{(elset_{i,k}-1)\cdot 2+1,1}$$

$d_{i-1,0} := i$

$$d = \begin{bmatrix} 1 & -1.293\cdot 10^{-3} & 3.113\cdot 10^{-3} & 1.327\cdot 10^{-4} & -1.101\cdot 10^{-4} & 3.723\cdot 10^{-8} & 2.651\cdot 10^{-4} \\ 2 & -1.849\cdot 10^{-3} & 6.276\cdot 10^{-3} & 2.331\cdot 10^{-4} & 7.037\cdot 10^{-5} & -1.289\cdot 10^{-5} & 8.92\cdot 10^{-5} \\ 3 & -2.928\cdot 10^{-3} & 0.012 & 4.122\cdot 10^{-4} & -4.706\cdot 10^{-6} & -1.289\cdot 10^{-5} & 4.475\cdot 10^{-4} \\ 4 & -5.051\cdot 10^{-3} & 0.015 & 6.372\cdot 10^{-4} & -1.248\cdot 10^{-4} & 1.187\cdot 10^{-4} & 7.428\cdot 10^{-5} \\ 5 & 2.788\cdot 10^{-4} & 3.038\cdot 10^{-3} & 1.294\cdot 10^{-4} & 2.368\cdot 10^{-5} & 1.795\cdot 10^{-8} & 2.589\cdot 10^{-4} \\ 6 & -2.775\cdot 10^{-4} & 6.201\cdot 10^{-3} & 2.314\cdot 10^{-4} & 8.59\cdot 10^{-5} & 3.217\cdot 10^{-5} & 3.619\cdot 10^{-5} \\ 7 & -1.357\cdot 10^{-3} & 0.012 & 4.106\cdot 10^{-4} & 1.082\cdot 10^{-5} & 3.217\cdot 10^{-5} & 3.945\cdot 10^{-4} \\ 8 & -1.239\cdot 10^{-3} & 0.015 & 3.978\cdot 10^{-4} & 5.191\cdot 10^{-5} & 6.595\cdot 10^{-5} & 3.158\cdot 10^{-4} \end{bmatrix}$$

$i := 1 .. ne$

$j := 0 .. 5$

$$\sigma_{i-1,j+1} := \sum_{k=0}^{5} E_{j,k} \cdot d_{i-1,k}$$

$\sigma_{i-1,0} := i$

$$\sigma = \begin{bmatrix} 1 & 0 & 0 & 0 & 4.381 & -3.091 & 5.728\cdot 10^{-4} \\ 2 & 0 & 0 & 0 & 11.172 & 6.167 & -0.198 \\ 3 & 0 & 0 & 0 & 18.058 & 5.229 & -0.198 \\ 4 & 0 & 0 & 0 & 26.365 & 2.919 & 1.826 \\ 5 & 0 & 0 & 0 & 6.002 & 2.748 & 2.762\cdot 10^{-4} \\ 6 & 0 & 0 & 0 & 11.304 & 6.827 & 0.495 \\ 7 & 0 & 0 & 0 & 18.189 & 5.89 & 0.495 \\ 8 & 0 & 0 & 0 & 18.172 & 7.528 & 1.015 \end{bmatrix}$$

==========

Appendix C

In this appendix, a further Mathcad Plus 6.0 [1995] file is reproduced. This file is designed to perform static, single time-step analyses on single blocks sub-meshed using 2nd order elements.

The example shown is a patch test. The geometry of the problem is shown in Figure C1. A -10% strain is applied in the x- and y-directions by prescribing displacements to the boundary nodes (see Section 1 of the file).

The file is a general file, and it can be used to solve other problems, by altering the data in Section 1.

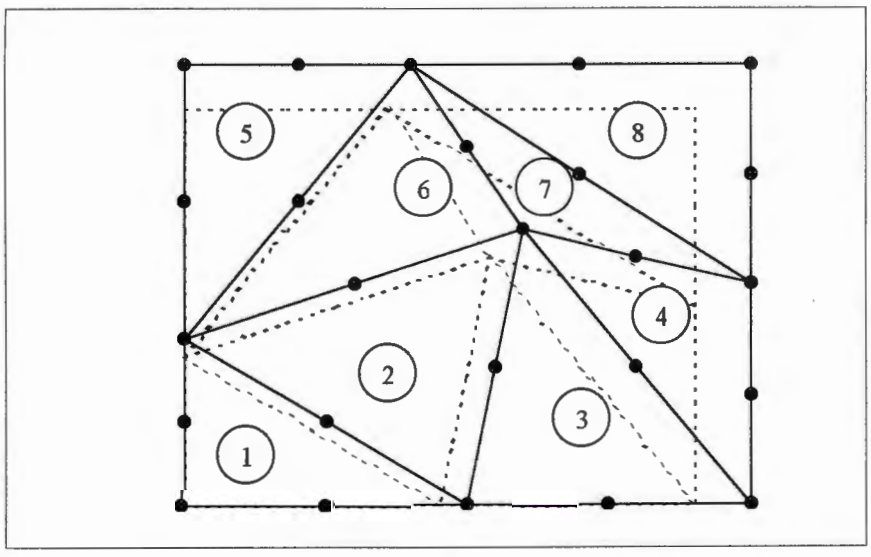


Figure C1: Meshing and displacement of the patch test.

MESHING: PATCH TEST USING 2nd ORDER ELEMENTS

1. User Input

$E := 40 \cdot 10^3$ (Young's modulus)

$v := 0.3$ (Poisson's ratio)

$p := 8000 \cdot 10^5$ (Penalty value)

$ne := 8$ (Number of elements)

$nn := 25$ (Number of nodes)

$nlp := 0$ (Number of load points)

$nfp := 15$ (Number of fixed points)

$nf2 := 1$ (Number of fixed points (2))

$rows := 5$

DATA ARRAY EXPLAINATIONS:

Node definition (nset):
Column 1: x co-ordinate
Column 2: y co-ordinate
Column 3: node number

Element definition (elset):
Columns 1..6: nodes 1..6
Column 4: element number

Fixed point definition:
Column 1: element number
Column 2: node number
Column 3: first d.o.f. in which point is fixed
Column 4: last d.o.f. in which point is fixed
Column 5: prescribed displacement, u_m
Column 6: prescribed displacement v_m

Load point definition:
Column 1: element number
Column 2: node number
Column 3: load in x direction
Column 4: load in y direction

$$nset := \begin{bmatrix} 0 & 0 & 0 \\ 0 & .8 & 1 \\ 0.2 & .8 & 2 \\ 0.4 & .8 & 3 \\ 0.7 & .8 & 4 \\ 1 & .8 & 5 \\ 0 & .55 & 6 \\ 0.2 & .55 & 7 \\ 0.5 & .65 & 8 \\ 0.7 & .6 & 9 \\ 1 & .6 & 10 \\ 0 & .3 & 11 \\ .3 & .4 & 12 \\ .6 & .5 & 13 \\ .8 & .45 & 14 \\ 1 & .4 & 15 \\ 0 & .15 & 16 \\ .25 & .15 & 17 \\ .55 & .25 & 18 \\ .75 & .2 & 19 \\ 1 & .2 & 20 \\ 0 & 0 & 21 \\ .25 & 0 & 22 \\ .5 & 0 & 23 \\ .75 & 0 & 24 \\ 1 & 0 & 25 \end{bmatrix}$$

$$elset := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 21 & 22 & 23 & 17 & 11 & 16 & 1 \\ 23 & 18 & 13 & 12 & 11 & 17 & 2 \\ 23 & 19 & 15 & 14 & 13 & 18 & 3 \\ 23 & 24 & 25 & 20 & 15 & 19 & 4 \\ 11 & 7 & 3 & 2 & 1 & 6 & 5 \\ 11 & 12 & 13 & 8 & 3 & 7 & 6 \\ 13 & 14 & 15 & 9 & 3 & 8 & 7 \\ 15 & 10 & 5 & 4 & 3 & 9 & 8 \end{bmatrix}$$

$$fp := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 2 & 0 & 0 \\ 4 & 3 & 1 & 2 & -.1 & 0 \\ 4 & 4 & 1 & 2 & -.1 & .02 \\ 4 & 5 & 1 & 2 & -.1 & .04 \\ 8 & 2 & 1 & 2 & -.1 & .06 \\ 8 & 3 & 1 & 2 & -.1 & .08 \\ 8 & 4 & 1 & 2 & -.07 & .08 \\ 8 & 5 & 1 & 2 & -.04 & .08 \\ 5 & 4 & 1 & 2 & -.02 & .08 \\ 5 & 5 & 1 & 2 & 0 & .08 \\ 1 & 6 & 1 & 1 & 0 & 0 \\ 1 & 5 & 1 & 1 & 0 & 0 \\ 5 & 6 & 1 & 1 & 0 & 0 \\ 1 & 2 & 2 & 2 & 0 & 0 \\ 1 & 3 & 2 & 2 & 0 & 0 \end{bmatrix}$$

$$lp := (0 \ 0 \ 0 \ 0)$$

$$fp2 := \begin{pmatrix} 0 & 0 & 2 & 2 & 0 & 0 \\ 4 & 2 & 2 & 2 & 0 & 0 \end{pmatrix}$$

2. Mesh Formulation

$i := 1 .. ne$

$x_{i,1} := nset_{(elset_{i,0}),0}$

$x_{i,2} := nset_{(elset_{i,1}),0}$

$x_{i,9} := nset_{(elset_{i,0}),1}$

$x_{i,10} := nset_{(elset_{i,1}),1}$

$x_{i,3} := nset_{(elset_{i,2}),0}$ $x_{i,11} := nset_{(elset_{i,2}),1}$

$x_{i,4} := nset_{(elset_{i,3}),0}$ $x_{i,12} := nset_{(elset_{i,3}),1}$

$x_{i,5} := nset_{(elset_{i,4}),0}$ $x_{i,13} := nset_{(elset_{i,4}),1}$

$x_{i,6} := nset_{(elset_{i,5}),0}$ $x_{i,14} := nset_{(elset_{i,5}),1}$

$x_{i,7} := x_{i,1}$ $x_{i,15} := x_{i,9}$

$i := 1 .. ne$

$$S_{i,0} := \frac{1}{2} \cdot \left[\sum_{k=1}^{6} (x_{i,k} \cdot x_{i,k+9} - x_{i,k+1} \cdot x_{i,k+8}) \right]$$

$$S_{i,1} := \frac{1}{6} \cdot \sum_{k=1}^{6} (x_{i,k} \cdot x_{i,k+9} - x_{i,k+1} \cdot x_{i,k+8}) \cdot (x_{i,k} + x_{i,k+1})$$

$$S_{i,2} := \frac{1}{6} \cdot \sum_{k=1}^{6} (x_{i,k} \cdot x_{i,k+9} - x_{i,k+1} \cdot x_{i,k+8}) \cdot (x_{i,k+8} + x_{i,k+9})$$

$x_{i,0} := \frac{S_{i,1}}{S_{i,0}}$ $x_{i,8} := \frac{S_{i,2}}{S_{i,0}}$

$i := 1 .. ne$

$j := 1 .. 7$

$xb_{i,j} := x_{i,j} - x_{i,0}$

$xb_{i,j+8} := x_{i,j+8} - x_{i,8}$

$$T(i,j) := \begin{bmatrix} 1 & 0 & -(xb_{i,j+8}) & (xb_{i,j}) & 0 & \frac{1}{2}\cdot(xb_{i,j+8}) & \frac{1}{2}\cdot(xb_{i,j})^2 & (xb_{i,j})\cdot(xb_{i,j+8}) & 0 \\ 0 & 1 & (xb_{i,j}) & 0 & (xb_{i,j+8}) & \frac{1}{2}\cdot(xb_{i,j}) & 0 & 0 & (xb_{i,j})\cdot(xb_{i,j+8}) \end{bmatrix}$$

$$\frac{1}{2}\cdot(x \qquad 0 \qquad \frac{1}{2}\cdot(xb_{i,j})\cdot(xb_{i,j+8}) \qquad \frac{1}{4}\cdot(xb_{i,j+8})^2$$

$$i,j+8)^2 \qquad \frac{1}{4}\cdot(xb_{i,j})^2 \qquad \frac{1}{2}\cdot(xb_{i,j})\cdot(xb_{i,j+8})$$

3. Simplexes

$i := 1 .. ne$

$$S_{i,0} := \frac{1}{2}\cdot\left[\sum_{k=1}^{6}(xb_{i,k}\cdot xb_{i,k+9} - xb_{i,k+1}\cdot xb_{i,k+8})\right]$$

$$S_{i,1} := \frac{1}{6}\cdot\sum_{k=1}^{6}(xb_{i,k}\cdot xb_{i,k+9} - xb_{i,k+1}\cdot xb_{i,k+8})\cdot(xb_{i,k} + xb_{i,k+1})$$

$$S_{i,2} := \frac{1}{6}\cdot\sum_{k=1}^{6}(xb_{i,k}\cdot xb_{i,k+9} - xb_{i,k+1}\cdot xb_{i,k+8})\cdot(xb_{i,k+8} + xb_{i,k+9})$$

$$S_{i,3} := \frac{1}{12}\cdot\sum_{k=1}^{6}(xb_{i,k}\cdot xb_{i,k+9} - xb_{i,k+1}\cdot xb_{i,k+8})\cdot\left[(xb_{i,k})^2 + xb_{i,k}\cdot xb_{i,k+1} + (xb_{i,k+1})^2\right]$$

$$S_{i,4} := \frac{1}{12}\cdot\sum_{k=1}^{6}(xb_{i,k}\cdot xb_{i,k+9} - xb_{i,k+1}\cdot xb_{i,k+8})\cdot\left[(xb_{i,k-8})^2 + xb_{i,k+8}\cdot xb_{i,k+9} + (xb_{i,k+9})^2\right]$$

$$S_{i,5} := \frac{1}{24} \cdot \sum_{k=1}^{6} (xb_{i,k} \cdot xb_{i,k+9} - xb_{i,k+1} \cdot xb_{i,k+8}) \cdot (2 \cdot xb_{i,k} \cdot xb_{i,k+8} + xb_{i,k} \cdot xb_{i,k+9} + xb_{i,k+1} \cdot xb_{i,k+8} +$$

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0.075 & 0 & 0 & 1.042 \cdot 10^{-3} & 3.75 \cdot 10^{-4} & -3.125 \cdot 10^{-4} \\ 0.14 & 0 & 0 & 2.411 \cdot 10^{-3} & 1.478 \cdot 10^{-3} & 7.778 \cdot 10^{-5} \\ 0.105 & 0 & 0 & 1.225 \cdot 10^{-3} & 1.225 \cdot 10^{-3} & 6.125 \cdot 10^{-4} \\ 0.1 & 0 & 0 & 1.389 \cdot 10^{-3} & 8.889 \cdot 10^{-4} & 5.556 \cdot 10^{-4} \\ 0.1 & 0 & 0 & 8.889 \cdot 10^{-4} & 1.389 \cdot 10^{-3} & 5.556 \cdot 10^{-4} \\ 0.11 & 0 & 0 & 1.711 \cdot 10^{-3} & 1.161 \cdot 10^{-3} & 7.944 \cdot 10^{-4} \\ 0.05 & 0 & 0 & 7.778 \cdot 10^{-4} & 3.611 \cdot 10^{-4} & -4.722 \cdot 10^{-4} \\ 0.12 & 0 & 0 & 2.4 \cdot 10^{-3} & 1.067 \cdot 10^{-3} & -8 \cdot 10^{-4} \end{bmatrix}$$

---

4. DDA Stiffness Matrix Formulation

4.1 Initialise

$m := ne \cdot 12 - 1$

$i := 1 .. m$

$j := 1 .. ne$

$k := 0 .. 11$

$K_{k,i} := 0$

$F_{k,j} := 0$

4.2 Elastic strain matrix

$$KP1(i) := \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ S_{i,0} & v \cdot S_{i,0} & 0 \\ v \cdot S_{i,0} & S_{i,0} & 0 \\ 0 & 0 & \frac{1-v}{2} \cdot S_{i,0} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \frac{E}{1-v^2}$$

$$KP2(i) := \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ S_{i,3} & S_{i,5} & v \cdot S_{i,3} & v \cdot S_{i,5} & 0 & 0 \\ S_{i,5} & S_{i,4} + \frac{1-v}{2} \cdot S_{i,3} & v \cdot S_{i,5} & v \cdot S_{i,4} & \frac{1-v}{2} \cdot S_{i,3} & \frac{1-v}{2} \cdot S_{i,5} \\ v \cdot S_{i,3} & v \cdot S_{i,5} & S_{i,3} + \frac{1-v}{2} \cdot S_{i,4} & S_{i,5} & \frac{1-v}{2} \cdot S_{i,5} & \frac{1-v}{2} \cdot S_{i,4} \\ v \cdot S_{i,5} & v \cdot S_{i,4} & S_{i,5} & S_{i,4} & 0 & 0 \\ 0 & \frac{1-v}{2} \cdot S_{i,3} & \frac{1-v}{2} \cdot S_{i,5} & 0 & \frac{1-v}{2} \cdot S_{i,3} & \frac{1-v}{2} \cdot S_{i,5} \\ 0 & \frac{1-v}{2} \cdot S_{i,5} & \frac{1-v}{2} \cdot S_{i,4} & 0 & \frac{1-v}{2} \cdot S_{i,5} & \frac{1-v}{2} \cdot S_{i,4} \end{bmatrix} \cdot \frac{E}{1-v^2}$$

$i := 1 .. ne$

$j := 3 .. 5$

$k := 0 .. 11$

$$K_{k,(i-1)\cdot 12+j} := K_{k,(i-1)\cdot 12+j} + KP1(i)_{k,j-3}$$

$i := 1 .. ne$

$j := 6 .. 11$

$k := 0 .. 11$

$$K_{k,(i-1)\cdot 12+j} := K_{k,(i-1)\cdot 12+j} + KP2(i)_{k,j-6}$$

4.3 Point load

$n := 1 .. nlp$

$i := 0 .. 11$

$j := 0 .. 1$

$$F_{i,(lp_{n,0})} := F_{i,(lp_{n,0})} + T(lp_{n,0}, lp_{n,1})_{j,i} \cdot lp_{n,j+2}$$

index out of bounds

4.4 Prescribed displacement

$n := 1 .. nfp$

$i := 0 .. 11$

$j := 0 .. 11$

$$K_{i,(fp_{n,0}-1)\cdot 12+j} := K_{i,(fp_{n,0}-1)\cdot 12+j} + p \cdot \sum_{k=fp_{n,2}-1}^{fp_{n,3}-1} T(fp_{n,0}, fp_{n,1})_{k,i} \cdot T(fp_{n,0}, fp_{n,1})_{k,j}$$

$$F_{i,fp_{n,0}} := F_{i,fp_{n,0}} + p \cdot \sum_{k=fp_{n,2}-1}^{fp_{n,3}-1} T(fp_{n,0}, fp_{n,1})_{k,i} \cdot fp_{n,k+4}$$

$n := 1 .. nf2$

$i := 0 .. 11$

$j := 0 .. 11$

$$K_{i,(fp2_{n,0}-1)\cdot 12+j} := K_{i,(fp2_{n,0}-1)\cdot 12+j} + p\cdot \sum_{k=fp2_{n,2}-1}^{fp2_{n,3}-1} T(fp2_{n,0}, fp2_{n,1})_{k,i}\cdot T(fp2_{n,0}, fp2_{n,1})_{k,}$$

$$F_{i,fp2_{n,0}} := F_{i,fp2_{n,0}} + p\cdot \sum_{k=fp2_{n,2}-1}^{fp2_{n,3}-1} T(fp2_{n,0}, fp2_{n,1})_{k,i}\cdot fp2_{n,k+4}$$

==========

5. Derive Q- Matrices

```
QI := for m ∈ 1..ne
        for i ∈ 0..1
          for j ∈ 0..11
            Q_{i,j} ← T(m,1)_{i,j}
            Q_{i+2,j} ← T(m,2)_{i,j}
            Q_{i+4,j} ← T(m,3)_{i,j}
            Q_{i+6,j} ← T(m,4)_{i,j}
            Q_{i+8,j} ← T(m,5)_{i,j}
            Q_{i+10,j} ← T(m,6)_{i,j}
        QIT ← Q^-1
        for i ∈ 0..11
          for j ∈ 0..11
            QI_{i,(m-1)·12+j} ← QIT_{i,j}
        QI
```

6. Transform K- and F-matrices into FE format

$m := 1 .. ne$

$i := 0 .. 11$

$j := 0 .. 11$

$$KT_{i,(m-1)\cdot 12+j} := \sum_{k=0}^{11}\left[\sum_{l=0}^{11}\left(QI_{k,(m-1)\cdot 12+i}\cdot K_{k,(m-1)\cdot 12+l}\cdot QI_{l,(m-1)\cdot 12+j}\right)\right]$$

$$FT_{i,m} := \sum_{j=0}^{11} QI_{j,(m-1)\cdot 12+i}\cdot F_{j,m}$$

---

7. Initialise and Assemble Global Stiffness Matrix

$i := 0 .. nn\cdot 2 - 1$

$j := 0 .. nn\cdot 2 - 1$

$KF_{i,j} := 0$ $\qquad$ $FF_i := 0$

$m := 1 .. ne$

$i := 0 .. 1$

$j := 0 .. 1$

$$KF_{\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j} + KT_{i,(m-1)\cdot 12+j}$$

$$KF_{\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j} + KT_{i,(m-1)\cdot 12+2+j}$$

$$KF_{\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j} + KT_{i,(m-1)\cdot 12+4+j}$$

$$KF_{\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j} + KT_{i,(m-1)\cdot 12+6+j}$$

$$KF_{\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j} + KT_{i,(m-1)\cdot 12+8+j}$$

$$KF_{\left[\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,0}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j} + KT_{i,(m-1)\cdot 12+10+j}$$

$$KF_{\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j} + KT_{i+2,(m-1)\cdot 12+j}$$

$$KF_{\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j} + KT_{i+2,(m-1)\cdot 12+2+j}$$

$$KF_{\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j} + KT_{i+2,(m-1)\cdot 12+4+j}$$

$$KF_{\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j} + KT_{i+2,(m-1)\cdot 12+6+j}$$

$$KF_{\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j} + KT_{i+2,(m-1)\cdot 12+8+j}$$

$$KF_{\left[\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,1}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j} + KT_{i+2,(m-1)\cdot 12+10+j}$$

$$KF_{\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j} + KT_{i+4,(m-1)\cdot 12+j}$$

$$KF_{\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j} + KT_{i+4,(m-1)\cdot 12+2+j}$$

$$KF_{\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j} + KT_{i+4,(m-1)\cdot 12+4+j}$$

$$KF_{\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j} + KT_{i+4,(m-1)\cdot 12+6+j}$$

$$KF_{\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j} + KT_{i+4,(m-1)\cdot 12+8+j}$$

$$KF_{\left[\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,2}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j} + KT_{i+4,(m-1)\cdot 12+10+j}$$

$$KF_{\left[\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j} + KT_{i+6,(m-1)\cdot 12+j}$$

$$KF_{\left[\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j} + KT_{i+6,(m-1)\cdot 12+2+j}$$

$$KF_{\left[\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j} + KT_{i+6,(m-1)\cdot 12+4+j}$$

$$KF_{\left[\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j} + KT_{i+6,(m-1)\cdot 12+6+j}$$

$$KF_{\left[\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j} + KT_{i+6,(m-1)\cdot 12+8+j}$$

$$KF_{\left[\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,3}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j} + KT_{i+6,(m-1)\cdot 12+10+j}$$

$$KF_{\left[\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j} + KT_{i+8,(m-1)\cdot 12+j}$$

$$KF_{\left[\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j} + KT_{i+8,(m-1)\cdot 12+2+j}$$

$$KF_{\left[\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j} + KT_{i+8,(m-1)\cdot 12+4+j}$$

$$KF_{\left[\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j} + KT_{i+8,(m-1)\cdot 12+6+j}$$

$$KF_{\left[\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j} + KT_{i+8,(m-1)\cdot 12+8+j}$$

$$KF_{\left[\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,4}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j} + KT_{i+8,(m-1)\cdot 12+10+j}$$

$$KF_{\left[\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,0}-1\right)\cdot 2+j} + KT_{i+10,(m-1)\cdot 12+j}$$

$$KF_{\left[\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,1}-1\right)\cdot 2+j} + KT_{i+10,(m-1)\cdot 12+2+j}$$

$$KF_{\left[\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,2}-1\right)\cdot 2+j} + KT_{i+10,(m-1)\cdot 12+4+j}$$

$$KF_{\left[\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,3}-1\right)\cdot 2+j} + KT_{i+10,(m-1)\cdot 12+6+j}$$

$$KF_{\left[\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,4}-1\right)\cdot 2+j} + KT_{i+10,(m-1)\cdot 12+8+j}$$

$$KF_{\left[\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j\right]} := KF_{\left(elset_{m,5}-1\right)\cdot 2+i,\left(elset_{m,5}-1\right)\cdot 2+j} + KT_{i+10,(m-1)\cdot 12+10+j}$$

$$FF_{\left(elset_{m,0}-1\right)\cdot 2+i} := FF_{\left(elset_{m,0}-1\right)\cdot 2+i} + FT_{i,m}$$

$$FF_{\left(elset_{m,1}-1\right)\cdot 2+i} := FF_{\left(elset_{m,1}-1\right)\cdot 2+i} + FT_{i+2,m}$$

$$FF_{\left(elset_{m,2}-1\right)\cdot 2+i} := FF_{\left(elset_{m,2}-1\right)\cdot 2+i} + FT_{i+4,m}$$

$$FF_{\left(elset_{m,3}-1\right)\cdot 2+i} := FF_{\left(elset_{m,3}-1\right)\cdot 2+i} + FT_{i+6,m}$$

$$FF_{\left(elset_{m,4}-1\right)\cdot 2+i} := FF_{\left(elset_{m,4}-1\right)\cdot 2+i} + FT_{i+8,m}$$

$$FF_{\left(elset_{m,5}-1\right)\cdot 2+i} := FF_{\left(elset_{m,5}-1\right)\cdot 2+i} + FT_{i+10,m}$$

8. Solve for Nodal Displacements

$KFI := KF^{-1}$

$ndof := nn \cdot 2 - 1$

$i := 0 .. ndof$

$$U_{i,1} := \sum_{j=0}^{ndof} KFI_{i,j} \cdot FF_j$$

$$U_{i,0} := \text{floor}\left(\frac{i}{2} + 1\right)$$

$U =$

	0	1
0	1	$-3.205 \cdot 10^{-7}$
1	1	0.08
2	2	-0.02
3	2	0.08
4	3	-0.04
5	3	0.08
6	4	-0.07
7	4	0.08
8	5	-0.1
9	5	0.08
10	6	$-1.282 \cdot 10^{-6}$
11	6	0.055
12	7	-0.02
13	7	0.055
14	8	-0.05
15	8	0.065

$i := 0 .. rows - 1$

$$j := 0 .. \frac{nn}{rows} - 1$$

$Ua_{2 \cdot i, j} := U_{(2 \cdot i \cdot rows + 2 \cdot j), 1}$ $\quad Ua_{2 \cdot i + 1, j} := U_{(2 \cdot i \cdot rows + 2 \cdot j + 1), 1}$

$$Ua = \begin{bmatrix} -3.205329 \cdot 10^{-7} & -0.02 & -0.04 & -0.07 & -0.1 \\ 0.08 & 0.079999 & 0.079999 & 0.079998 & 0.08 \\ -1.281918 \cdot 10^{-6} & -0.020001 & -0.05 & -0.07 & -0.099999 \\ 0.055 & 0.054999 & 0.064999 & 0.059999 & 0.06 \\ -5.129767 \cdot 10^{-7} & -0.03 & -0.06 & -0.08 & -0.099999 \\ 0.03 & 0.04 & 0.05 & 0.045 & 0.04 \\ -7.690105 \cdot 10^{-7} & -0.025001 & -0.055 & -0.074999 & -0.099999 \\ 0.015001 & 0.015001 & 0.025 & 0.020001 & 0.02 \\ -1.924103 \cdot 10^{-7} & -0.025001 & -0.05 & -0.074999 & -0.1 \\ 3.205477 \cdot 10^{-7} & 1.28195 \cdot 10^{-6} & 6.410618 \cdot 10^{-7} & 1.281908 \cdot 10^{-6} & 3.205328 \cdot 10^{-7} \end{bmatrix}$$

(alternative form of displaying results)

9. Transform Back to DDA Block Unknowns

$i := 1 .. ne$

$j := 0 .. 11$

$$d_{i-1,j+1} := \sum_{k=0}^{5} QI_{j,(i-1)\cdot 12+2\cdot k} \cdot U_{(elset_{i,k}-1)\cdot 2,1} + \sum_{k=0}^{5} QI_{j,(i-1)\cdot 12+2\cdot k+1} \cdot U_{(elset_{i,k}-1)\cdot 2+1,1}$$

$d_{i-1,0} := i$

$d =$

	0	1	2	3	4	5	6
0	1	-0.017	0.01	$3.634 \cdot 10^{-7}$	-0.1	0.1	$1.419 \cdot 10^{-6}$
1	2	-0.037	0.027	$-8.068 \cdot 10^{-7}$	-0.1	0.1	$-7.842 \cdot 10^{-7}$
2	3	-0.07	0.03	$1.357 \cdot 10^{-6}$	-0.1	0.1	$7.701 \cdot 10^{-9}$
3	4	-0.083	0.013	$-1.223 \cdot 10^{-6}$	-0.1	0.1	$-1.698 \cdot 10^{-6}$
4	5	-0.013	0.063	$-1.441 \cdot 10^{-6}$	-0.1	0.1	$1.586 \cdot 10^{-6}$
5	6	-0.033	0.053	$2.234 \cdot 10^{-7}$	-0.1	0.1	$-4.796 \cdot 10^{-7}$
6	7	-0.067	0.057	$-2.052 \cdot 10^{-6}$	-0.1	0.1	$4.022 \cdot 10^{-7}$
7	8	-0.08	0.067	$2.141 \cdot 10^{-6}$	-0.1	0.1	$7.454 \cdot 10^{-7}$

oo00oo