UNIVERSITEIT • STELLENBOSCH • UNIVERSITY

An Explicit Finite Difference Method for Analyzing Hazardous Rock Mass

by

Gysbert Basson

Thesis presented in partial fulfilment of the requirements for the degree of Master of Sciences in Applied Mathematics at Stellenbosch University

Applied Mathematics Division, Department of Mathematical Sciences
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Supervisor: Dr. Francois Smit

December 2011

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: .....

Abstract

An Explicit Finite Difference Method for Analyzing Hazardous Rock Mass

Applied Mathematics Division, Department of Mathematical Sciences
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Thesis: MSc Applied Mathematics

December 2011

FLAC3D is a three-dimensional explicit finite difference program for solving a variety of solid mechanics problems, both linear and non-linear. The development of the algorithm and its initial implementation were performed by Itasca Consulting Group Inc. The main idea of the algorithm is to discritise the domain of interest into a Lagrangian grid where each cell represents an element of the material. Each cell can then deform according to a prescribed stress/strain law together with the equations of motion. An in-depth study of the algorithm was performed and implemented in Java. During the implementation, it was observed that the type of boundary conditions typically used has a major influence on the accuracy of the results, especially when boundaries are close to regions with large stress variations, such as in mining excavations. To improve the accuracy of the algorithm, a new type of boundary condition was developed where the FLAC3D domain is embedded in a linear elastic material, named the Boundary Node Shell (BNS). Using the BNS shows a significant improvement in results close to excavations. The FLAC algorithm is also quite amendable to paralellization and a multi-threaded version that makes use of

multiple Central Processing Unit (CPU) cores was developed to optimize the speed of the algorithm. The final outcome is new non-commercial Java source code (JFLAC) which includes the Boundary Node Shell (BNS) and shared memory parallelism over and above the basic FLAC3D algorithm.

Uittreksel

An Explicit Finite Difference Method for Analyzing Hazardous Rock Mass

Applied Mathematics Division, Department of Mathematical Sciences
University of Stellenbosch
Private Bag X1, 7602 Matieland, South Africa

Tesis: MSc Applied Mathematics

Desember 2011

FLAC3D is 'n eksplisiete "eindige verskil" program wat 'n verskeidenheid liniêre en nie-liniêre soliede meganika probleme kan oplos. Die oorspronklike algoritme en die implimentasies daarvan was deur Itasca Consulting Group Inc. toegepas. Die hoofidee van die algoritme is om 'n gebied te diskritiseer deur gebruik te maak van 'n Lagrangese rooster, waar elke sel van die rooster 'n element van die rooster materiaal beskryf. Elke sel kan dan vervorm volgens 'n sekere spannings/vervormings wet. 'n Indiepte ondersoek van die algoritme was uitgevoer en in Java geïmplimenteer. Tydens die implementering was dit waargeneem dat die grense van die rooster 'n groot invloed het op die akkuraatheid van die resultate. Dit het veral voorgekom in areas waar stress konsentrasies hoog is, gewoonlik naby areas waar myn uitgrawings gemaak is. Dit het die ontwikkelling van 'n nuwe tipe rand kondisie tot gevolg gehad, sodat die akkuraatheid van die resultate kon verbeter. Die nuwe rand kondisie, genaamd die Grens Node Omhulsel (GNO), aanvaar dat die gebied omring is deur 'n elastiese materiaal, wat veroorsaak dat die grense van die gebied 'n elastiese reaksie het op die stress binne die gebied. Die GNO het 'n aansienlike

verbetering in die resultate getoon, veral in areas naby myn uitgrawings. Daar was ook waargeneem dat die FLAC algoritme parralleliseerbaar is en het gelei tot die implentering van 'n multi-SVE weergawe van die sagteware om die spoed van die algoritme te optimeer. Die finale uitkomste is 'n nuwe nie-kommersiële Java weergawe van die algoritme (JFLAC), wat die implimentering van die nuwe GNO randwaardekondisie insluit, asook toelaat vir die gebruik van multi- Sentrale Verwerkings Eenheid (SVE) as 'n verbetering op die basiese FLAC3D algoritme.

Acknowledgments

Foremost, I would like to thank the Institute of Mining Seismology (IMS) for the support of this thesis, especially Alexander Mendecki and Richard Lynch who envisions IMS as leaders in integrated seismicity. A special thanks goes to my colleagues and friends: Renoir Sewjee, for everything he taught me about numerical modelling and his support during the study, and Assen Ilchev, for sharing his brilliance in mathematics and physics during the development of the Boundary Node Shell.

Thank you Francois for having so much patience with me, especially with regards to my bad grammar. Thank you for always treating me with respect and never loosing your temper when I repeatedly made mistakes.

A very special thanks goes to my parents, Bertie and Riana Basson, who always believed in me, and for always encouraging me to be the best. Lastly, the two girls in my life, my wife Ilza and daughter Rebecca, deserve all the credit and also deserves this degree. Thank you Ilza, for supporting me during this part time study. Thank you for looking after our baby, and keeping her quiet when I had to work late nights.

Dedications

This thesis is dedicated to my wife, Ilza, and our beautiful baby girl, Rebecca.

Contents

List of Figures

List of Tables

List of Abbreviations

BEM	Boundary Element Method
BC	Bushveld Complex
BNS	Boundary Node Shell
BZ	Bullnose
CFL	Courant–Friedrichs–Lewy condition
CPU	Central Processing Unit
DGD	Dual Grid Discritization
FEM	Finite Element Method
FLAC	Fast Lagrangian Analysis of Continua
FLAC3D	3D commertial software version of FLAC
JFLAC	Java version of FLAC compiled by the author of this study
LSM	Large Strain Mode
MD	Mixed Discritization
MND	Mixed Nodal Discritization
MPM	Material Point Method
PIC	Particle In Cell Method

SSM	Small Strain Mode
UCS	Uni-axial Compressive Strength

List of Symbols

A^l	A surface of a tetrahedron
a	Area
$\mathbf{a}$, a_i	Acceleration
B_i	Body force components used in dynamic equation of motion
$\mathbf{b}$, b_i	Body force components in static equation of motion
C	Courant number
C_0	Cohesion
$\mathbf{E}$	Strain matrix
E	Young's modulus
$F_i^{d^{N^z}}$	Damping force componenets of global node N^z
$F_i^{N^l}$	Force components acting on node N^l
$F_i^{N^z}$	Force components acting on global node N^z
F^s	Shear failure condition
F^t	Tensile failure condition
$\mathbf{f}$	Force
f_x	Force component in the $x-$direction of a Cartesian coordinate system
G	Shear modulus
g	Flow rule
g^s	Shear flow rule
g^t	Tensile flow rule

$\mathbf{I}$	Identity matrix
K	Bulk modulus
L	Length
N^l	A node of a tetrahedron
N^z	A global node in the system
M^{N^z}	Mass of global node N^z
m^{N^l}	Mass of node N^l
$\mathbf{n}$	Normal vector to a particular surface
n	Total number of nodes in the system
n_b	Total number of boundary nodes
$\mathbf{n}^{A^k}$	Normal of a tetrahedron surface
$p_i^{(k)}$	Reference points of the Boundary Node Shell
q	Symbol used to indicate a uniform stress field
R	Radius
R_f	Radius of fracture zone
$\mathbf{r}$, r_i	Position
$r_i^{N^l}$	Position of node N^l
S	Closed surface contour
s_c	Actual support resistance
$s_c^{'}$	Effective support resistance
$s_i^{(k)}$	Source points of the Boundary Node Shell
s_{sign}	Symbol used to represent either -1 or 1
T	Stress matrix
$T_{ij}(s_i^{(k)}, p_i^{(m)}, \mathbf{n}_i^{(m)})$	Kelvin traction kernel
T^h	A tetrahedron in the system
t	Time
$\triangle t$	Time step

$\mathbf{t}$, t_i Traction

$U_{ij}(s_i^{(k)}, p_i^{(m)})$ Kelvin displacement kernel

$\mathbf{u}$, u_i Displacement

$\nabla\mathbf{u}$, $u_{i,j}$ Displacement gradient matrix

V Volume

V^{T^h} Volume of a tetrahedron

$\mathbf{v}$, v_i Velocity

$v_i^{A^k}$ Velocity componentsof a tetrahedron surface

$\overline{v}_i^{A^k}$ Mean velocity components of a tetrahedron surface

$\nabla\mathbf{v}$, $v_{i,j}$ Velocity gradient matrix

$\triangle v_i$ Velocity increment

v_{max} Maximum pressure wave velocity

$\mathbf{W}$ Rotation matrix

W^{ext} External work

W^{int} Internal work

w Number of tetrahedrons sharing a node

w^{ext} External work-rate

w^{int} Internal work-rate

$\triangle x$ Smallest distance between two grid nodes

$\mathbf{y}$ Centroid of a tetrahedron

Greek Symbols

α_1, α_2 Material constants

β Constant used in the definition of a homogeneous function

γ Damping constant

δ_{ij} Kronecker delta

ϵ, ϵ_{ij} Strain / Strain components

$\dot{\epsilon}$, $\dot{\epsilon}_{ij}$	Strain / Strain-rate components
$\dot{\bar{\epsilon}}$	Average strain-rate
$\dot{\epsilon}_{ij}^{T^h}$	Strain-rate of a tetrahedron
ϵ_1	Major principal strain
ϵ_2	Principal strain component
ϵ_3	Minor principal strain
$\triangle\epsilon_{ij}$	Strain increment
$\triangle\epsilon_{ij}^e$	Elastic strain increment
$\triangle\epsilon_{ij}^p$	Plastic strain increment
$\triangle\epsilon_{ij}^{p^{T^h}}$	Plastic strain increment of a tetrahedron
$\dot{\zeta}_{ij}^{T^h}$	Deviatoric component of the strain-rate tensor of a tetrahedron
η	Constant depending on spatial coordinates
λ	Scalar value used in eigenvalue decomposition
λ_L	Lamé constant
ν	Poisson's ratio
ρ	Material density
σ, σ_{ij}	Stress / Stress components
σ_1	Major principal stress
σ_2	Principal stress component
σ_3	Minor principal stress
σ_c	Uni-axial compressive strength
σ_v	Virgin stress
σ_{ij}^I	Elastic guess of the stress
σ_{ij}^N	Newly calculated stress
$\sigma_{ij}^{n^{T^h}}$	Newly calculated stress after application of MND
$\sigma_{ij}^{o^{T^h}}$	Old stress value before application of MND
σ_{ij}^T	Total stress

$\triangle\sigma$, $\triangle\sigma_{ij}$ Stress increment

$\triangle\sigma^{p^{T^h}}$ Plastic stress increment of a tetrahedron

σ_n Normal stress

σ_t Tension cut-off used in the Mohr-Coulomb criterion

σ_{xx} Component of the stress matrix

τ Shear stress

ϕ Friction angle

ψ Dilatation angle

ω_{ij} Rotation tensor

$\dot{\omega}_{ij}$ Rotation-rate tensor

Miscellaneous

$< N^a >$ Computational list of all grid nodes in the system

$< T^k >$ Computational list of all the tetrahedrons in the system

map Computational connectivity map

Chapter 1

Introduction

1.1 Background

Earthquakes remain one of the most dangerous natural disasters, claiming many lives through history. An earthquake, also known as a tremor or seismic event, is an event of sudden failure of a part of the earth's lithosphere that radiates seismic waves. Most earthquakes result from slip along existing faults under tectonic stress. Faults are planar discontinuities along which parts of the rockmass have slipped past each other. A dyke is another type of geological discontinuity along which earthquakes may originate. A dyke can be thought of as a steeply dipping fault with infilling igneous rock, having been the conduit of molten rock en route towards the earth's surface, where it may have flowed out as lava. Where the country rock is weaker than the igneous infilling, the dyke can become a stress concentrator when the rock mass is deformed.

The amplitudes and frequencies of seismic waves radiated by an earthquake are measured by various instruments that measure ground acceleration, velocity or displacement. From the strength of the recorded ground motions and knowledge of distance from the source, the strength of the earthquake can be estimated, usually represented by the Richter magnitude [16]. The Richter magnitude is based on the logarithm of the maximum amplitde of ground motion and can range from negative values to the maximum ever recorded being 9.5.

Small magnitude earthquakes occur regularly around the globe, but every few months a large magnitude earthquake occurs somewhere in the world. The San Andreas fault zone in central California is well known for causing large earthquakes. A considerable amount of research has been done on this fault zone to allow improved insight into all aspects of the earthquake phenomenon. The scientific literature on earthquakes is vast.

Earthquakes also occur in the mining environment. When rock masses are removed from an ore body, stresses can accumulate on the surrounding geological structures. Once the stress levels become too high to sustain, the structure slips and seismic waves are emitted. The ground motions caused by these seismic events, can loosen already fractured rocks and fatal accidents can result.

Several methods have been developed to estimate the hazard of rock mass in underground mines. Information provided by historical seismic events is used to measure the state of the rock mass; location, moment and the radiated energy of these seismic events are used to calculate parameters such as Energy Index [22], Schmidt Number[14], Apparent Stress Level or Cumulative Seismic Displacement [13]. These parameters are analyzed on a daily basis and if one of these parameters exceeds a given threshold, it could indicate the possibility of a large seismic event. A sudden increase in the frequency of seismic activity in a concentrated area could also be an indication of a larger seismic event. The above mentioned techniques are all examples of useful early warning systems, but the major issue of the exact date and time of the next major seismic event still remains unpredictable.

1.2 Aim

This investigation attempts to aid scientists in determining the potential hazard of a volume of rock in the mining environment. The approach selected for this study is to develop a computer generated stress model for a particular mining configuration. In situ stresses that are physically measured underground, are used together with the current mining configuration as input to the simulation. The final result is an estimation of the

stresses inside the rock mass. In most cases the stresses under which certain rock types fail are known from laboratory strength testing. If the stress model indicates that the investigated rock mass has stresses close to these failure strengths, then it is possible that these highly stressed areas could fail. It is important to note that this technique is not a prediction system for seismic events but rather a tool for better understanding of the environment and possibly indicating hazardous areas.

Several methods exist to model complex underground excavations. The Boundary Element Method (BEM) [2] is well-known for its ability so simulate stress levels for large mining configurations. Several domain methods, such as the Finite Element Method (FEM) [20] can also accomplish this task. The Material Point Method (MPM) [21] is an extension of the Particle-In-Cell (PIC) Method [8], commonly used in computational fluid dynamics, to solid dynamics and is capable of simulating stresses inside the domain as well as to identify possible material failure regions in the domain. In the MPM, the domain is discritised into Lagrangian point masses, or material points, that moves through a Eulerian background mesh.

The above techniques, and others, have both advantages and disadvantages. The BEM has the advantage of solving large scale models in a reasonable time frame. Its disadvantage is that it is limited to an isotropic elastic medium that assumes that no failure can occur in the body. FEM has been applied successfully to a wide range of problems with good results. However, three-dimensional objects can be difficult and time consuming to implement in body fixed FEM meshes. Furthermore, solution accuracy is compromised when large deformations are present in FEM simulations. The deformations lead to mesh distortion and usually requires re-meshing the domain, which again may be time consuming. The MPM is capable of simulating large deformations, without remeshing the domain. However, its is computationally more expensive than FEM in terms of storage, since information about material points and the background has to be stored. Also, particles may oscillate if it crosses boundaries of the background mesh.

The modeling and simulation method that will be focused on in this study was deve-

loped by Itasca Consulting Group Inc. and is known as *Fast Lagrangian Analysis of Continua, (FLAC)*. FLAC is an explicit finite difference algorithm for solving a variety of solid mechanics problems. This method uses basic constitutive equations to define the material, and the algorithm uses a set of partial differential equations derived from general principles to relate the mechanical (stress) and kinematic (strain rate, velocity) variables. The principles include the definition of strain and laws of motion. These differential equations are solved for a particular geometry where the user defines the material properties and initial conditions.

FLAC is not restricted to underground geometry, but for the use of this study, it will only be applied to the mining environment. This method of modelling rock mass is not necessarily better than FEM codes, but is much simpler to implement and does not involve solving the complex FEM equations. However, as in FEM, the problem of remeshing arise when large deformations become evident in the domain. Also, modelling complex geometries may be a difficult and time consuming task. Although FLAC has these disadvantages, it still remains one of the most popular modelling tool in the mining industry.

Two-dimensional and three-dimensional versions of FLAC exist. For the purpose of this study, the 3D version will be discussed in detail and all references to FLAC refers to this. The documentation that describes the basic FLAC algorithm [10], that is supplied with a copy of the FLAC software, was used as the basis for the development of a FLAC version in Java. For future reference, this Java Code will be referred to as JFLAC.

During the implementation of the algorithm, it was observed that the type of boundary conditions typically used in FLAC, have a significant influence on the accuracy of the simulated results. This became more evident as boundaries were placed closer to areas where results were analyzed. This led to the implementation of a new type of boundary condition, called the Boundary Node Shell (BNS), which is an addition to FLAC. The BNS assumes that the entire domain is placed in a linear elastic material and the boundaries of the modelled domain responded elastically to the contained body forces. This showed a significant improvement in simulation accuracy. An additional contribution was

made by creating a multi- Central Processing Unit (CPU) version of the basic algorithm. The final outcome is new non-commercial Java source code (JFLAC) which includes the Boundary Node Shell (BNS) and shared memory parallelism over and above the basic FLAC algorithm.

1.3 Layout of this document

Chapter 1: Introduction

This chapter introduces the reader to the problem and gives a motivation behind the work that is focused on in this study. A brief background on seismic hazard in rock mass is given and it mentions different tools that can be used to measure the hazard. The aim of the study is given and a motivation for choosing FLAC as a modelling hazard analysis tool is also discussed. Lastly, a few improvements to the original FLAC algorithm that were developed and implemented is addressed.

Chapter 2: Governing equations

This chapter discusses the basic governing equations, such as stress, strain and the equations of motion, used in the JFLAC algorithm. A general discussion on the inelastic response in soil, by means of the Mohr-Coulomb condition, is also given.

Chapter 3: The JFLAC algorithm

The nodal formulations of the governing equations are derived. A detailed description of the JFLAC grid is given and it is explained how the domain of interest is discritised into a Lagrangian grid. Consequently the derivations of the basic algorithm and its implementation are described.

Chapter 4: Contributions to JFLAC

This chapter describes the contributions the author of this study made to the basic FLAC algorithm. The implementation of the new type of boundary condition, the Boundary

Node Shell (BNS), is described. The development of a multi-threaded version of the JFLAC algorithm is also discussed.

Chapter 5: Results and discussion

JFLAC is tested against a well known analytical solution. It is also compared with the results obtained by FLAC3D for the same problem. A case study is then performed on a South-African mine in the Bushveld Complex. The performance of the BNS is also tested.

Chapter 6: Summary and conclusion

This chapter summarizes the work and achievements of the study and provides several conclusions.

Chapter 7: Possible improvements

Possible improvements to the JFLAC algorithm are given here.

Chapter 2

Governing equations

This chapter reviews the basic governing equations that are used in JFLAC. These equations include the basic elements of stress and strain as well as some fundamentals of elasticity. It also presents a general discussion of inelastic response in soil by means of the Mohr-Coulomb condition and provides a brief explanation of inelastic flow. This chapter is fundamental to the following chapters and follows Davis and Selvadurai [5].

2.1 Deformation and strain

An important aspect of a solid body is the description of its deformations. The term "deformation" refers to the motion of any particular particle in the body as well as the overall motion of the body and is usually the result of external forces that act on the body. To better illustrate this, consider the elastic body A in Fig. (2.1) that has a reference configuration A_0. In this configuration the body is free from any load. If a set of external forces is applied to the body in its reference configuration, it will change to its deformed configuration, denoted by A_t, where the subscript t refers to time. A displacement vector $\mathbf{u}_t$ can be introduced that connects the position of a particular particle from its reference configuration to its deformed configuration. If a displacement vector is drawn for every particle in A from A_0 to A_t, a vector field may be formed and can be written as

$$\mathbf{u} = \mathbf{u}(\mathbf{r}, t), \tag{2.1}$$

where $\mathbf{r}$ is the position vector of all the particles in A_0. In linear elasticity theory it is assumed that deformations are small, such that the position of a particle in its reference configuration and deformed configuration are for all practical purposes, identical.

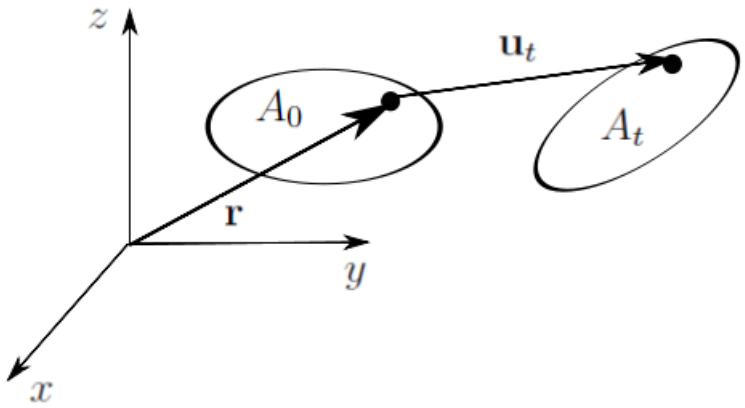


Figure 2.1: The reference configuration A_0 and deformed configuration A_t for a generic elastic body.

While deformations often lead to strains within a body, it is important to note that not all deformations will lead to strain. The deformations that do not strain a body consist of either *rigid translations* or *rigid rotations*. A rigid translation is any deformation that does not depend on $\mathbf{r}$. This implies that if $\mathbf{u}$ is the same for every $\mathbf{r}$, the body is undergoing a rigid translation. If the body is rotated around a fixed axis, then it is undergoing a rigid rotation. The major difference between straining and the two rigid motions described is that only strains change the shape and/or length of the body.

Strain may be defined as the change in length ($\triangle L$) of a deformed body, normalized with respect to the original undeformed length (L), and is mathematically expressed as

$$\epsilon = \frac{\triangle L}{L}. \tag{2.2}$$

This form of strain is known as extensional strain.

Only strains will result in stresses within a body. To characterize these stresses, the

strains must first be determined, which can only be achieved after all rigid body motions are eliminated. Firstly, a distinction between rigid translations and strains can be made by analyzing the variation of the vector field $\mathbf{u}$ around a single point in a body. The partial derivative of $\mathbf{u}$ is taken for this point in a rectangular Cartesian coordinate system. This is known as the displacement gradient matrix and is expressed as

$$\nabla \mathbf{u} = \begin{bmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_x}{\partial y} & \frac{\partial u_x}{\partial z} \\ \frac{\partial u_y}{\partial x} & \frac{\partial u_y}{\partial y} & \frac{\partial u_y}{\partial z} \\ \frac{\partial u_z}{\partial x} & \frac{\partial u_z}{\partial y} & \frac{\partial u_z}{\partial z} \end{bmatrix}. \tag{2.3}$$

The partial derivatives in Eq. (2.3) will not be affected by rigid translations, since all the derivatives will be zero. This might suggest that Eq. (2.3) can be used as a measure of strain. However, rigid rotations would, in some cases, lead to non-zero derivatives. To make distinction between rigid rotations and strains,$\nabla \mathbf{u}$ is further refined by decomposing it into two matrices, one being symmetric and the other skew-symmetric. The symmetric matrix is called the strain matrix, $\mathbf{E}$, and is defined by

$$\mathbf{E} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T), \tag{2.4}$$

where T denotes the transpose of the matrix. The skew-symmetric matrix is called the rotation matrix, $\mathbf{W}$, and is defined by

$$\mathbf{W} = \frac{1}{2}(\nabla \mathbf{u} - \nabla \mathbf{u}^T). \tag{2.5}$$

To follow the convention used throughout this study, Eqs. (2.4) and (2.5) are used in their tensor forms. This is done by adding an indicial notation to present the dependent and independent variables in Eq. (2.1). The position vector $\mathbf{r}$ can be denoted by r_i where i can take on the values 1, 2, 3. Consequently Eq. (2.1) can be expressed as

$$u_i = u_i(r_i, t). \tag{2.6}$$

Following the tensor notation of Flügge [7], the displacement gradient matrix then be-

comes:

$$u_{i,j} = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} \\ u_{2,1} & u_{2,2} & u_{2,3} \\ u_{3,1} & u_{3,2} & u_{3,3} \end{bmatrix} \tag{2.7}$$

where indices i and j follow the Cartesian tensor notation. Consequently Eqs. (2.4) and (2.5) become

$$\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i}) \tag{2.8}$$

and

$$\omega_{ij} = \frac{1}{2}(u_{i,j} - u_{j,i}), \tag{2.9}$$

respectively. Further, by differentiating Eqs. (2.8) and (2.9) with respect to time yields

$$\dot{\epsilon}_{ij} = \frac{1}{2}(v_{i,j} + v_{j,i}) \tag{2.10}$$

and

$$\dot{\omega}_{ij} = \frac{1}{2}(v_{i,j} - v_{j,i}) \tag{2.11}$$

for the strain-rate- and rotation-rate tensors respectively, where v_i are velocity componenets.

2.2 Stress

2.2.1 Elasticity

Stresses will develop in a material if a body is strained. The concept of stress can be understood as the force acting on some surface area in the body. For example, if a cross-section is made perpendicular to a rope that is under tension, then the traction vector, $\mathbf{t}$, can be defined as the force in the rope divided by the cross-sectional area of the rope. Mathematically this is expressed as

$$\mathbf{t} = \frac{\textbf{force vector}}{area}. \tag{2.12}$$

Although the described concept is easy to understand, it is not as straightforward to define **t** for cross-sections that are not perpendicular to the rope. It necessitates comprehensively describing the stress in the rope. Cauchy overcame the challenge by showing how to find the traction on any surface through the rope, by looking at tractions on three specific surfaces. To illustrate this, consider a 3D generic body as shown in Fig. (2.2) [5].

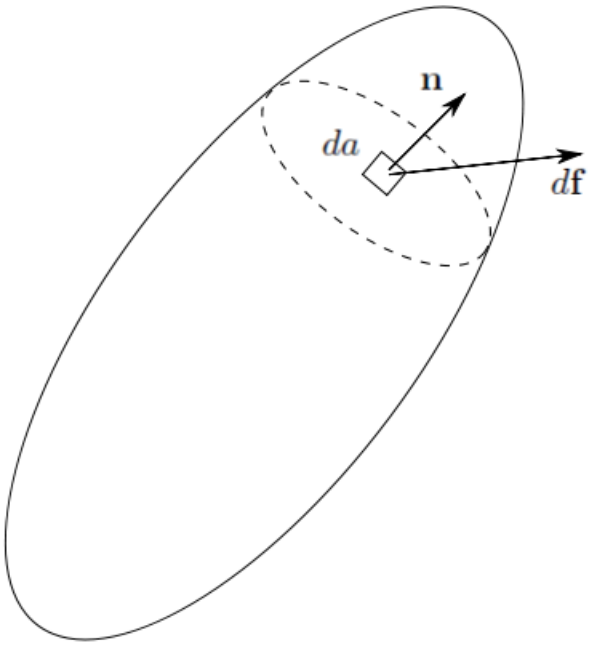


Figure 2.2: 3D generic shaped figure for describing the traction vector.

A cross-section is made through the body and the area element, da, will be analyzed. Let $d\mathbf{f}$ be the force that acts on da. The traction vector can be defined by the limit:

$$\mathbf{t} = \lim_{da \to 0} \frac{d\mathbf{f}}{da}. \tag{2.13}$$

Next define **n** as the unit vector that is normal to da. Cauchy showed that the traction vector can be determined by taking the product of a 3x3 matrix with the normal vector

$\mathbf{n}$, expressed as

$$\mathbf{t} = \mathbf{T} \cdot \mathbf{n} \tag{2.14}$$

where $\mathbf{T}$ is known as the stress matrix. This matrix contains all the information needed to find the traction on any surface that passes through the body. The expression in Eq. (2.14) gives the components of $\mathbf{t}$ in three coordinate directions. However, in some cases it is useful to look at the components of $\mathbf{t}$ that acts normal and tangential to a surface. The vector component that is perpendicular to the surface is referred to as the normal stress, σ_n, and can be calculated by

$$\sigma_n = \mathbf{t} \cdot \mathbf{n}. \tag{2.15}$$

The tangential component, referred to as the shear stress τ, can be calculated by [5]

$$\tau = \sqrt{\mathbf{t} \cdot \mathbf{t} - \sigma_n^2}. \tag{2.16}$$

A familiar way to obtain a graphical representation of the stress state at a point in terms of τ and σ_n in a stressed medium, is the Mohr diagram [9]. If τ and σ_n are analyzed for any point in the body, and every possible orientation of surface that passes through the point is considered, it is found that all values for τ and σ_n lie within a well-defined region. This region is shown in Fig. (2.3). Different stress states will have different Mohr diagrams with circles of different sizes. Often the outer most points where the circle intersects the normal stress axis are of interest. These points of intersection are known as the *principal stresses* and are the surfaces that intersects the points where $\mathbf{t}$ is parallel to the normal vector $\mathbf{n}$, i.e.

$$\mathbf{t} = \lambda \mathbf{n}, \tag{2.17}$$

where λ is a scalar value.

The principal stresses can be determined by substituting Eq. (2.17) into Eq. (2.14), yielding:

$$\lambda \mathbf{n} = \mathbf{T}\mathbf{n}. \tag{2.18}$$

Eq. (2.18) can be rearranged to become

$$(\mathbf{T} - \lambda\mathbf{I})\mathbf{n} = \mathbf{0}, \tag{2.19}$$

where $\mathbf{I}$ denotes the identity matrix. Eq. (2.19) is identified as an eigenvalue problem. The eigenvalues and eigenvectors can be determined by solving [5]

$$\det(\mathbf{T} - \lambda\mathbf{I}) = 0. \tag{2.20}$$

The eigenvalues for Eq. (2.20) are known as the principal stresses and are denoted by σ_1, σ_2 and σ_3 respectively. The corresponding eigenvectors define the three principal surfaces. For convenience, the principal stresses are usually numbered such that $\sigma_1 \geq \sigma_2 \geq \sigma_3$.

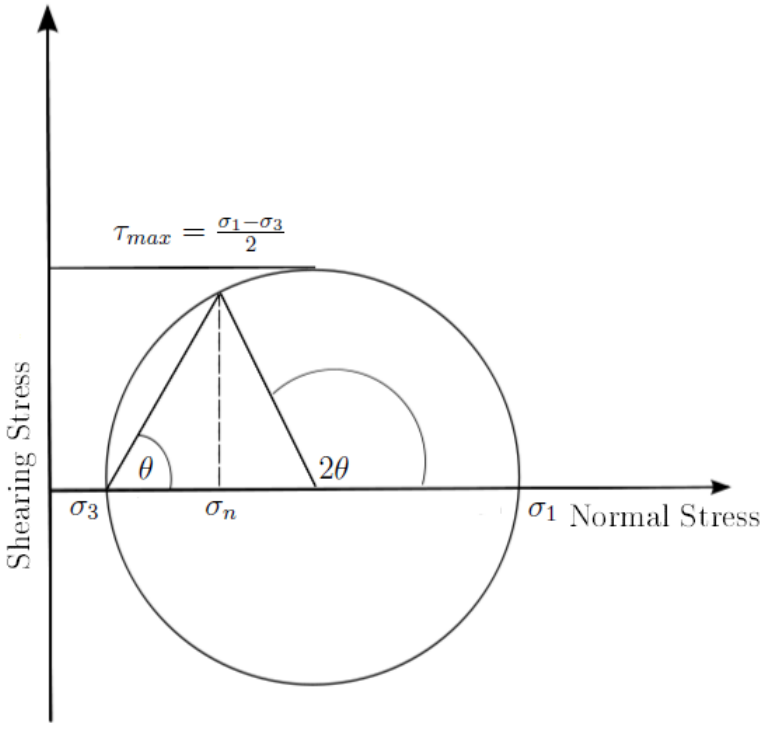


Figure 2.3: The Mohr-circle for visualizing principal stresses.

Now that the concepts of strain and stress are defined, there should exist a relation

between the strain at a point in a deformed body to the stress of this particular point. Hooke's law provides this relation and is the cornerstone in the theory of elasticity. This linear relationship between stress and strain is expressed as [5]

$$\mathbf{T} = \mathbf{CE} \tag{2.21}$$

where $\mathbf{C}$ is a matrix containing properties of the material. Since strain is a dimensionless quantity, $\mathbf{C}$ should have the same dimensions of stress.

A familiar quantity in linear elasticity is a material property called Young's modulus, represented as E. A second familiar quantity of elasticity is the Poisson's ratio, ν. To get a better understanding of these quantities, consider an elastic cylindrical bar that is subject to tension. The extension of the cylinder can be described in terms of the principal strain, ϵ_1, and Young's modulus provides the relationship to calculate the stress, σ_1. The cylinder also suffers a lateral contraction, ϵ_2 , as a result of the longitudinal extension. The ratio of lateral contraction as a result of longitudinal extension is known as Poisson's ratio, and is dimensionless. Mathematically, Poisson's ratio is expressed as

$$\nu = \frac{\epsilon_2}{\epsilon_1}. \tag{2.22}$$

If a shear load τ is applied to the cylinder, the material will experience a shearing strain which is directly proportional to the applied stress and a shear modulus, G. Like Young's modulus, G also has a dimension of stress and a relation can be expressed as [5]

$$G = \frac{E}{2(1+\nu)}. \tag{2.23}$$

Another modulus that gives the relationship between elastic volume change and stress is called the Bulk modulus, K, and contains information about the compressibility of the material. An expression for K is

$$K = \lambda_L + \frac{2}{3}G, \tag{2.24}$$

where λ_L is known as the Lamé material constant and is also expressed as

$$\lambda_L = \frac{\nu E}{(1+\nu)(1-2\nu)}. \tag{2.25}$$

The Possion ratio can also be expressed in terms of the Lamé constant by

$$\nu = \frac{\lambda_L}{2(\lambda_L + G)}. \tag{2.26}$$

2.2.2 Plasticity

In the theory of elasticity, total reversibility in the state of deformation is assumed for any elastic material. This implies that the material can obtain an infinitely large load without experiencing any damage, and once the load is removed from the material, it will return to its original state. However, in reality, this is not physical. All materials will reach a point where reversibility is lost and becomes permanently deformed, or even breaks, if a large enough load is applied. Fig. (2.4) shows a universal stress-strain curve [17] that highlights the relation between a material's elastic region and the plastic region, where the reversibility of the material is lost.

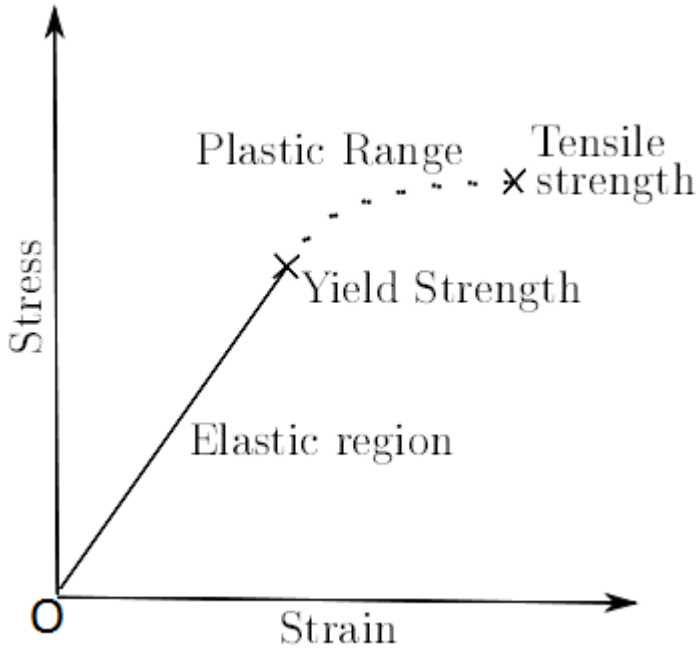


Figure 2.4: The stress-strain curve of a material.

Consider a cylindrical bar that is under no strain. Since the cylinder is unstrained, the stress-strain relation inside the cylinder can be placed at origin point, O, on the stress-strain curve in Fig. (2.4). Let a slowly increasing uni-axial load be applied to the cylinder. As the load increases, the stress-strain relation of the cylinder falls within the elastic region, and the stress-strain relation obeys Hooke's law. As the load increases, the stress-strain relation reaches a point known as the material's *yield strength*. At this point the material falls within the plastic range and the material becomes permanently deformed. This implies, that if the load is suddenly removed from the cylinder, the stress-strain relation of the material will not return back to the origin point O on the curve, but will return to a different state that does not lie on the curve. If the load continues to increase, it will reach a point, known as the material's *tensile strength,* where the material breaks.

A material is seen as plastic if it behaves elastically when stresses in the material are

below a yield strength, but when the applied stress is higher than the critical level, it flows continuously without rupture and becomes permanently deformed. Many attempts have been made to determine a common yield stress for all materials, but it was discovered that there are as infinitely many yield strengths as there are materials. Two conditions used for the mathematical handling of yield strength of metals are those of H. Tresca and R. von Mises [9]. The Mohr-Coulomb condition is well known when dealing with soils and rocks.

Coulomb [9] found that the strength of materials could be derived from its cohesion, i.e. the ability of particles in the material to stick together, represented as C_0, and its angle of internal friction ϕ, i.e. the critical angle at which a load must be applied to a material to fail under shear. His observations revealed that failure in soils could usually be associated with a surface of rupture. Restricting his attention to the surface, he wrote the failure criterion as:

$$\tau = C_0 + \sigma_n \tan\phi. \tag{2.27}$$

The shear stress limit in Eq. (2.27) provides the yield limit at which a material starts to behave plastically. The trend of Eq. (2.27) is a straight line and can be plotted on the Mohr diagram of a two-dimensional stress state in $\tau - \sigma$ space as illustrated in Fig. (2.5). If, for a given stress state, failure occurs, the combination of principal stresses must be tangent to this line. Therefore values of τ and σ can be related to the principal stresses σ_1 and σ_3. From Fig. (2.5) it can be shown that the shear stress value at which failure occurs is

$$\tau = \frac{1}{2}(\sigma_1 - \sigma_3)\cos\phi \tag{2.28}$$

and the corresponding normal stress is calculated from

$$\sigma_n = \frac{1}{2}(\sigma_1 + \sigma_3) - \frac{1}{2}(\sigma_1 - \sigma_3)\sin\phi. \tag{2.29}$$

By substituting Eq. (2.29) into Eq. (2.27), the Mohr-Coulomb condition can also be expressed as

$$\tau = \frac{1}{2}(\sigma_1 + \sigma_3)\sin\phi + C_0\cos\phi. \tag{2.30}$$

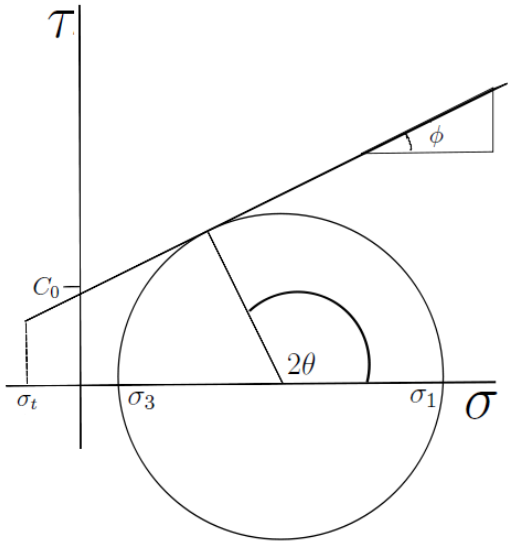


Figure 2.5: Mohr-Coulomb failure criterion on a $\tau - \sigma$ axis.

The Mohr-Coulomb condition assumes that a material only fails under shear. However a tension cutoff, σ_t, can be introduced into this model. Fig. (2.5) shows a tension cut-off before the straight line of Eq. (2.27) intersects the σ axis. If the tension exceeds this limit the material is assumed to fail under tension.

The Mohr-Coulomb failure envelope can also be shown in principal stress space ($\sigma_1 - \sigma_3$ space) as illustrated in Fig. (2.6). It can also be shown that by mathematical manipulation of Eqs. (2.27), (2.28) and (2.29), the linear relation found in Eq. (2.27), translates into

$$\sigma_1 = \sigma_c + N_\phi \sigma_3, \tag{2.31}$$

where

$$N_\phi = \frac{1 + \sin\phi}{1 - \sin\phi} \tag{2.32}$$

and the Uniaxial Compressive Strength (UCS - the capacity of a material or structure to withstand axially directed compressing forces) given by

$$\sigma_c = 2C_0\sqrt{N_\phi} \tag{2.33}$$

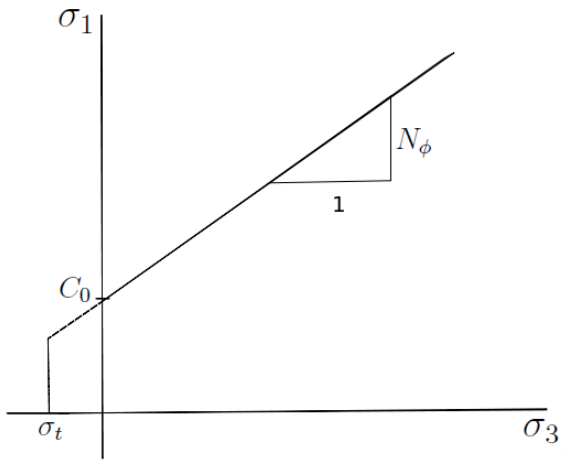


Figure 2.6: Coulomb failure criterion on a $\sigma_1 - \sigma_3$ axis.

The Mohr-Coulomb condition can also be expressed as a function of the stress by substituting Eqs. (2.28) and (2.29) into (2.27) [9], yielding

$$F(\sigma_1, \sigma_3) = \frac{\sigma_1 + \sigma_3}{2} \sin\phi - \frac{\sigma_1 - \sigma_3}{2} - C_0 \cos\phi. \tag{2.34}$$

This form of the Mohr-Coulomb failure criterion is applicable to failure on a plane parallel to the σ_2 plane. If Eq. (2.34) is extended to three-dimensions, then the resulting surface is a cone with a hexagonal cross section as illustrated in Fig. (2.7).

The stress state of any point inside a material can be analyzed by the surface in Fig. (2.7). If the principal stress values for this stress state is such that $F(\sigma_1, \sigma_3) < 0$, then the material is in its elastic range. If the values of these principle stresses are such that $F(\sigma_1, \sigma_3) = 0$, then the material is considered to be in the elastic-plastic range. Values for the principle stresses that causes $F(\sigma_1, \sigma_3) > 0$ are not allowed and a correction must be made to return it to the surface of the cone. This is usually done by applying a flow rule.

Details of the flow rules that are applied in JFLAC are described in Section (3.4). In brief, two types of flow rules exist, namely associated flow and non-associated flow. When

a material's friction angle is the same as its dilation angle (deformation angle, ψ), an associated flow can be applied to return the material back to the surface of the cones. This means that if a material is deformed by a stress, it will return to its original undeformed state if the stress is removed. However, if the dilation angle differs from the friction angle, the material will not completely return to its original state if the stress is removed, but will have some form of resulting deformation. In this case, a non-associated flow rule applies.

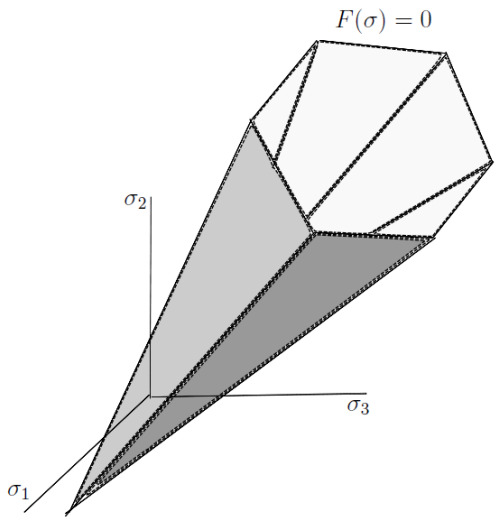


Figure 2.7: Mohr-Coulomb yield surface in $\sigma_1 : \sigma_2 : \sigma_3$ space.

2.3 The equilibrium equations

All forces on a body can be classified into two categories: namely contact forces and body forces. Contact forces are associated with surfaces and generally they lead to tractions as discussed in Section (2.2). Body forces are associated with volumes inside the body. Examples of body forces are gravitational and magnetic forces.

In Fig. (2.8) the stresses acting on an infinitesimal small cubular element, taken from a

stressed body in static equilibrium, are shown. Following [5], if the forces are in equilibrium, then it follows from Fig. (2.8) [5], in the x direction, that

$$(\sigma_{xx} + \frac{\partial \sigma_{xx}}{\partial x} dx) dydz - \sigma_{xx} dydz \quad + \tag{2.35}$$

$$(\sigma_{xy} + \frac{\partial \sigma_{xy}}{\partial_y} dy) dxdz - \sigma_{xy} dxdz \quad + \tag{2.36}$$

$$(\sigma_{xz} + \frac{\partial \sigma_{xz}}{\partial_y} dz) dxdy - \sigma_{xz} dxdy \quad + \tag{2.37}$$

$$\rho f_x dxdydz \quad = \quad 0. \tag{2.38}$$

In Eqs. (2.35) to (2.38) $dxdydz$ is the volume of the element and $\rho f_x dxdydz$ is the total body force in the x-direction, where ρ is known as the material density. Quantities $dydz$ and $dxdz$ are the areas of the cube faces. By combining these quantities Eqs. (2.35) to (2.38) becomes

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{xy}}{\partial y} + \frac{\partial \sigma_{xz}}{\partial z} + \rho f_x = 0. \tag{2.39}$$

Components for the y- and z directions are similarly computed.

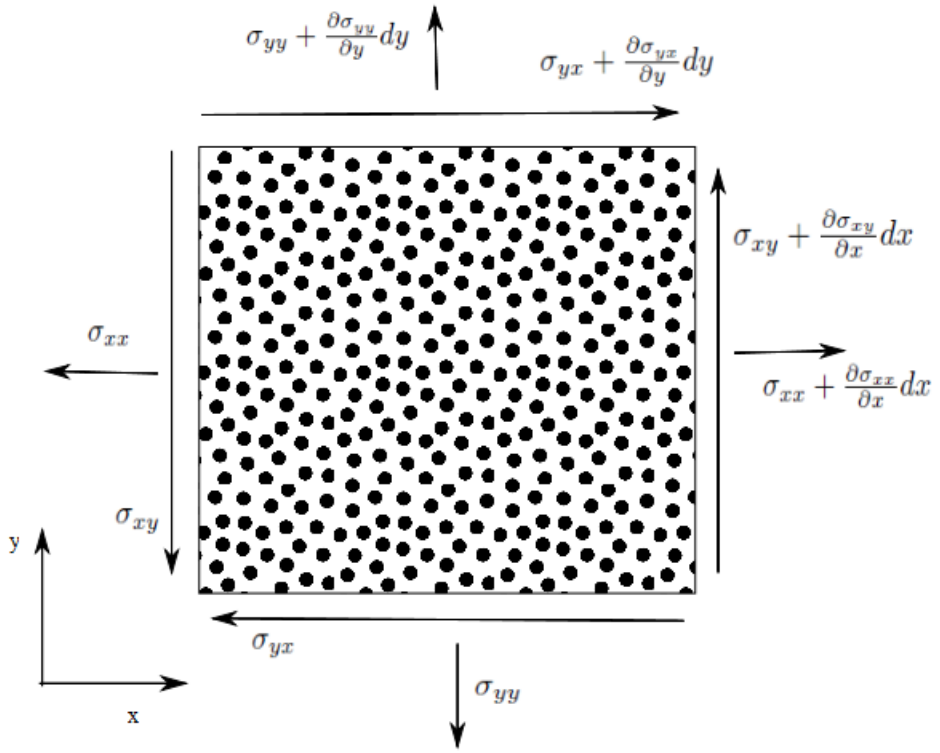


Figure 2.8: Stresses acting on a volume element [5].

The following three equations, that involve partial derivatives of the stress components, describe the concept of static equilibrium:

$$\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \sigma_{yx}}{\partial y} + \frac{\partial \sigma_{zx}}{\partial z} + \rho f_x = 0, \tag{2.40}$$

$$\frac{\partial \sigma_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \sigma_{zy}}{\partial z} + \rho f_y = 0 \tag{2.41}$$

and

$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + \rho f_z = 0 \tag{2.42}$$

where f_x, f_y and f_z are the components of the body force. These equations must hold for every point in the body as long as its remains in static equilibrium. Equations (2.40) to

(2.42) can be expressed in tensor form as

$$\sigma_{ij,j} + \rho b_i = 0, \tag{2.43}$$

and gives the condition of equilibrium for the forces acting on a volume element, where σ_{ij} is known as the stress tensor. If the body from which the volume element was taken had some initial motion, it implies that the body is in dynamic equilibrium. The dynamic equation of motion, also known as Cauchy's equation of motion, for the volume element [6] is then expressed as

$$\sigma_{ij,j} + \rho b_i = \rho \frac{dv_i}{dt}, \tag{2.44}$$

where the term on the right hand side is introduced to represent the initial motion of the body. In the computational simulation of dynamic problems (bodies moving through space), a reference system, also known as a grid, is adopted through which this body moves. This reference system must then obey one of the following formulations:

1. In the Lagrangian formulation (adopted in JFLAC) coordinates of the reference system are *attached* to the individual points of the body. These coordinates are allowed to move and deform together with the body. Using this formulation, the constitutive equations are used to determine what happens to *a specific point of the body*.

2. The Eulerian formulation uses a *fixed, rigid coordinate system* with the body moving relative to points of this fixed coordinate system. In this formulation the constitutive equations are used to determine what happens to *a specific point in space*.

For small motions of a solid, both of the above formulations coincide.

2.4 Summary

The basic theory behind the governing equations used in JFLAC were discussed. A description of deformation and strain were given. The relation between stress and strain in the form of Hooke's law were described, and the Mohr-Coulomb condition was used as a means to describe failure in a solid. The basic equations of motion were also covered.

Chapter 3

The JFLAC algorithm

This chapter describes the derivation of the governing equations used in JFLAC. The formulations derived in Chapter 2 are used as a basis to derive the nodal formulations of the governing equations that JFLAC inherited from FLAC. Any contributions that were added to the JFLAC algorithm are described in Chapter 4. This chapter follows some of the derivations contained in the FLAC verification manual [10].

3.1 Defining the grid

A 3D geometric model in JFLAC is by default discritised into hexahedral zones, called cells. The vertices of each cell, also known as nodes, form a Lagrangian grid that deforms with the material. Each cell is then internally discritised into sets of tetrahedron elements. The elements can deform according to prescribed stress/strain laws together with the equations of motion. Hexahedron elements are not used internally because of the possibility of hour-glassing: a common problem that occurs when hexahedrons deform in such a way that their corner vertices get close to opposite faces. The hexahedron looses its square shape and may collapse under high stresses.

JFLAC can also use a "tetrahedron only" model as input, but this model has to be well defined since tetrahedrons that have unusually sharp edges, can cause problems. An

example of such a problem is when a node of the tetrahedron penetrates its opposite face. This results into the tetrahedron obtaining a negative volume.

FLAC uses FISH (a programing language compiled by ITASCA) to generate models. When FISH is used to generate a model, internal algorithms are used to reduce the probability of obtaining irregular shaped elements. For the purpose of this study, all the models were generated using custom Java code that was developed during the implementation of JFLAC.

The grid generation procedure is explained in this section. Furthermore, the term *tetrahedron* used in this document refers to a tetrahedron element of the JFLAC domain. Similarly, the term *hexahedron* refers to a hexahedron element, or zone, of the JFLAC domain.

3.1.1 Discretising the domain

Consider a cube, also known as a hexahedron zone in JFLAC, with 8 vertices as illustrated in Fig. (3.1). A model consists of a number of these zones that are all connected. The zones do not need to be the same size, and side lengths can vary along the x, y and z axes in a Cartesian coordinate system. Generally, the zones are discritised finely in areas where high levels of accuracy are needed (normally close to an excavation in the mining environment) and more coarsely when the grid is a suitable distance away from these areas. Ideally a user would like to discritise the entire volume into very small zones, but due to computational limits, the number of zones needs to be carefully managed as the model can easily become very large and requires lots of computer resources to execute. Even with the current technology available, it might still be impossible to run in some cases. Hence keeping the number of zones at an optimal level is very important, for accuracy, as well as for the time needed to execute such a simulation.

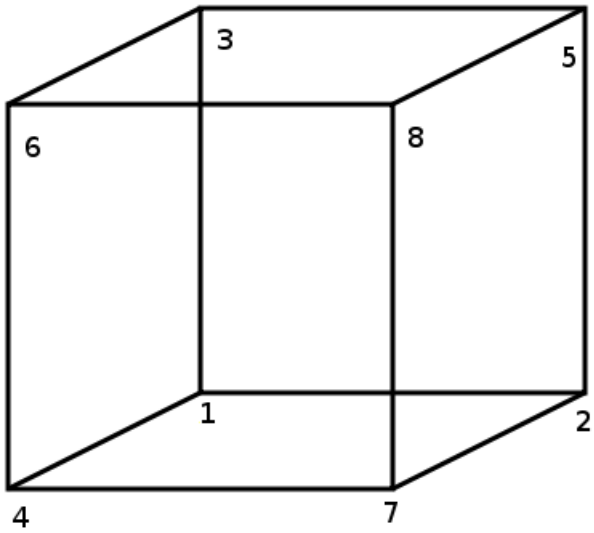
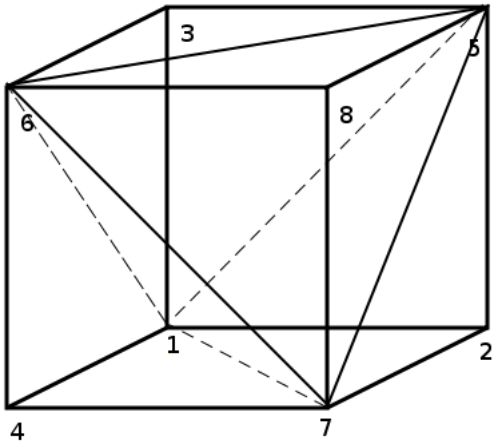


Figure 3.1: 8 Vertices of a cube.

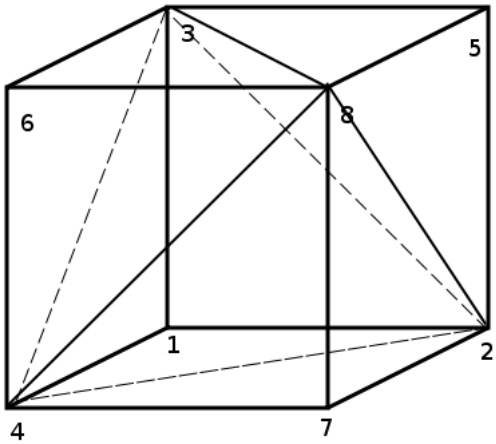
The numbering of the nodes in Fig. (3.1) is according to the right hand rule of numbering, to ensure that the normal of each face of the zone always points in an outward direction away from the center of the zone. Once the user has generated a set of zones to simulate the model, it is used as input to the JFLAC algorithm. Each of these zones are then internally split in one of two ways into a set of five tetrahedrons. The two ways, or overlays, of zone Discritisation into tetrahedrons are shown in Fig. (3.2). Overlay 1 in Fig. (3.2a) illustrates one way to split the zone into a set of 5 tetrahedrons. The connectivity of the 5 tetrahedrons using Overlay 1 is as follows:

1. Tetrahedron (a) - {4,6,7,1}
2. Tetrahedron (b) - {6,7,8,5}
3. Tetrahedron (c) - {1,7,2,5}
4. Tetrahedron (d) - {1,6,5,3}
5. Tetrahedron (e) - {1,6,5,7}

A zone can similarly be split into a set of tetrahedrons using Overlay 2, as shown in Fig. (3.2b).



(a) Splitting a hexahedron zone into tetrahedrons using Overlay 1.



(b) Splitting a hexahedron zone into tetrahedrons using Overlay 2**.**

Figure 3.2: Splitting a hexahedron zone into tetrahedrons.

Each tetrahedron has four nodes and four faces. Once a set of tetrahedrons is obtained using either of the overlays, then the faces of each tetrahedron need to be identified and

a normal unit vector (that points in an outwardly direction, i.e. away from the tetrahedron centroid) is generated for each face. This makes it easy to identify the inside of the tetrahedron.

3.1.2 The steps involved in creating a JFLAC model

Generating a model that can be used as input in a JFLAC simulation can be a complex process. A simple two-dimensional rectangular mining excavation will be used as an illustration to the steps involved in the creation of this model. In 2D, hexahedrons reduce to quadrilaterals and tetrahedrons to triangles. This will make visualization of the process easier and more understandable.

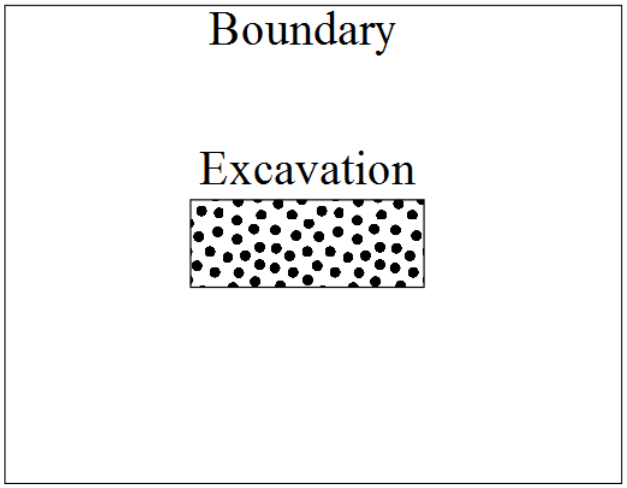
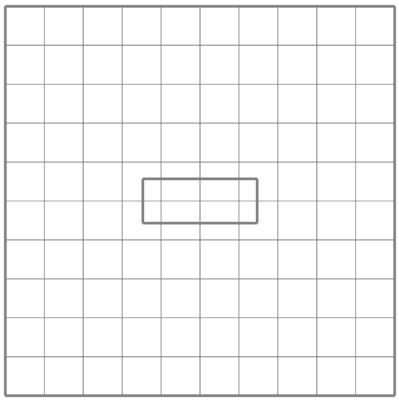


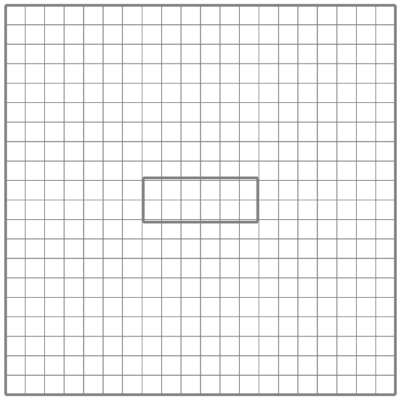
Figure 3.3: 2D rectangular mining excavation.

Consider the excavation as illustrated in Fig. (3.3). The first step involved in generating the model is to discritise the domain in Fig. (3.3) into quadrilateral cells. As previously mentioned, the size of these cells play a vital role in the simulation accuracy of the model as well as the computational power required to execute the simulation. To illustrate this, the domain will be discritised into a set of larger cells and smaller cells as

shown in Fig. (3.4a) and Fig. (3.4b) respectively. For now it will be assumed that the boundaries of the domain are a sufficient distance away from the excavation such that boundary effects do not become evident in the results. Boundary value problems that can occur in the JFLAC simulation are discussed in Section (3.2).



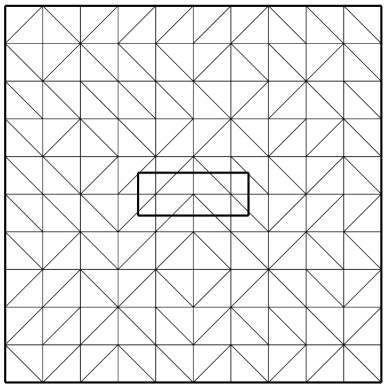
(a) JFLAC domain discritised into larger cells.



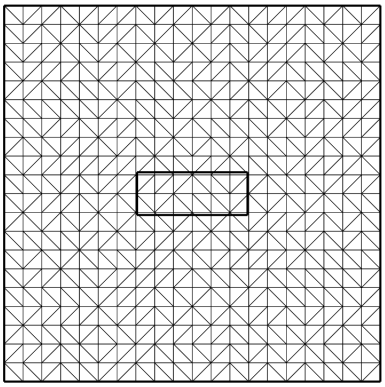
(b) JFLAC domain discritised into smaller cells.

Figure 3.4: JFLAC domain discritisation into cells.

Once the domain has been discritised into quadrilateral cells, then each cell is discritised into a set of two triangles as shown in Fig. (3.5). The choice of triangular discritisation is done randomly to avoid introducing artificial anisotropy.



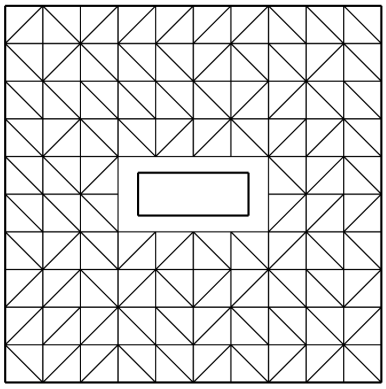
(a) Random discritising of larger cells into triangles.



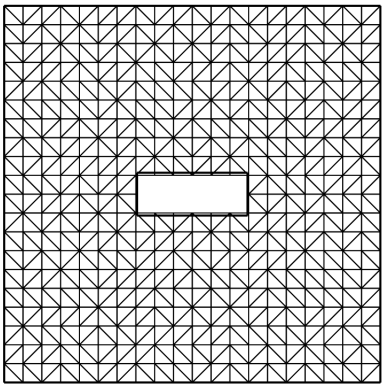
(b) Random discritising smaller cells into triangles.

Figure 3.5: Discritising cells into triangles.

Lastly, triangles that fall within the excavation in Fig. (3.5) are removed from the model to give the final model output shown in Fig. (3.6).



(a) Removal of triangles that fall within the excavation of the larger cells.



(b) Removal of triangles that fall within the excavation of the smaller cells.

Figure 3.6: Removal of triangles that fall within the excavation.

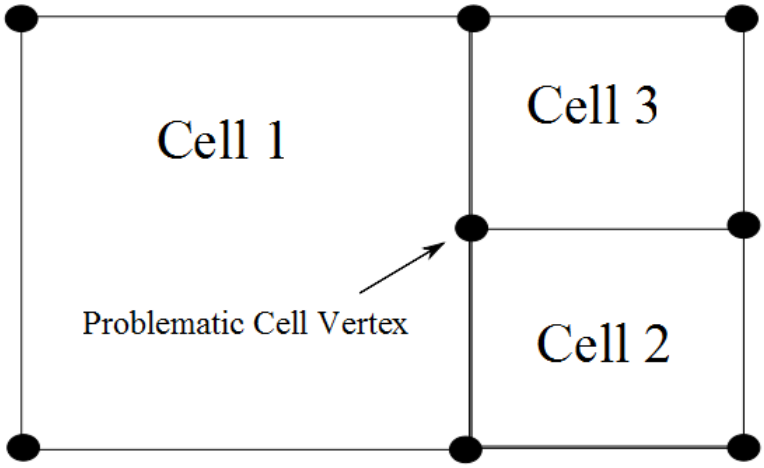


Figure 3.7: Example configuration of cells that is not allowed in a JFLAC simulation.

Note that the number of triangles shown in Fig. (3.6a) is significantly less than the number of triangles in Fig. (3.6b). This will lead to much faster simulation time, but since the excavation is overestimated, accuracy of the results can be disputed. The model containing smaller cells in Fig. (3.6b), gives a more accurate representation of the domain. The results will also be more accurate. However, simulation time can increase significantly.

Constant cell sizes, as used in this illustration, are not a prerequisite for a JFLAC model. Cells varying in size may also be used as input as long as each vertex, also known as a grid node, of a cell is connected to the vertices of its neighboring cells. Fig. (3.7) illustrates a configuration of cells that is not allowed, i.e. where for example vertices of Cell 2 and Cell 3 connect on the face of Cell 1.

Variable size meshes are generally more complicated to create and since it is not the focus of this study, regular grids were used in all of the simulations. There are more advanced mesh generation tools available that can be used to keep the number of cells in the model to a minimum as well as to increase model accuracy.

3.1.2.1 Duel Grid Discritization

Duel Grid Discritization (DGD) is the process where all the zones in a JFLAC domain are discritised into tetrahedrons by using both of the overlays in Fig. (3.2). Usually all zones are internally discritised into tetrahedrons by using only one of the overlays. But when the user specifies that DGD must be applied to the zones, the program internally discritise the zones into tetrahedrons using both overlays. Each zone then contains 10 tetrahedrons instead of 5.

The overlapping tetrahedrons do not cause a problem in the simulation, since tetrahedrons are only used to store the mechanical state of the system for each time step. The equations of motion are solved at the grid nodes for each time step and these results are used to determine whether the system is in an equilibrium state.

DGD is usually applied to a zone if high stress gradients exist between the tetrahedrons inside the zone. As mentioned earlier, hour-glassing is a problem that can occur in hexahedrons. This problem can be overcome by discritising the hexahedron into tetrahedrons using either of the overlays. But this does not always solve the problem. Tetrahedrons do not contain enough modes of deformation to fully allow for all the possible deformations the hexahedron can have. During the testing phase of JFLAC, it was found that zones that are close to excavated parts in the domain still deformed in such a way that the tetrahedrons became irregularly shaped. This problem was solved by implementing DGD.

It appears that by discritising a zone using only one overlay, anisotropy is introduced into the system. And if all the zones of the domain are discritised by using only one overlay, this problem becomes worse. When a zone is discritised using both overlays, anisotropy is avoided and the system is more stable.

Although DGD creates stability in the system, is has one major drawback. The total number of tetrahedrons in the system is double the number if only one overlay is used. This leads to almost double the amount of computational memory needed to solve this system. This can also increase simulation time significantly. It is possible to apply DGD

to only specified zones in the system, but a special algorithm is needed to identify these zones. This is however not implemented in JFLAC.

3.2 Boundary conditions

Boundary conditions play a vital role in simulation accuracy. Defining the correct boundary conditions is a fine art and it could take some time to find the best placement of the boundaries to obtain the best results. Having a simulation model with boundaries that are very close to an area where results with high accuracy is needed, cannot be trusted. Boundary effects can become noticeable in the results. However if the boundary is far from the excavation, the simulation model becomes large and it might increase the simulation time.

Following on the previous section, Section (3.1), the vertices, or nodes, of all the tetrahedrons that define the grid, are identified. Nodes that lie inside the domain are labeled as internal nodes whereas nodes that lie on the boundary of the domain are labeled as boundary nodes. A boundary condition is placed on *all* nodes. Internal nodes usually receive a free boundary condition, meaning the nodes can move and deform freely. The boundary nodes can also be free, but as least some boundary nodes must have a condition other than free. If all boundary nodes were assumed to be free, the simulation model will relax such that the contained body forces become zero.

There are three types of boundary conditions used in FLAC. The first being the free boundary as already discussed. The other two boundaries are known as displacement boundaries and stress boundaries. A fourth type of boundary condition known as the "Boundary Node Shell" (BNS) boundary condition, was introduced in JFLAC. This boundary condition is described in detail in Chapter 4. The different boundary conditions are illustrated in Fig. (3.8).

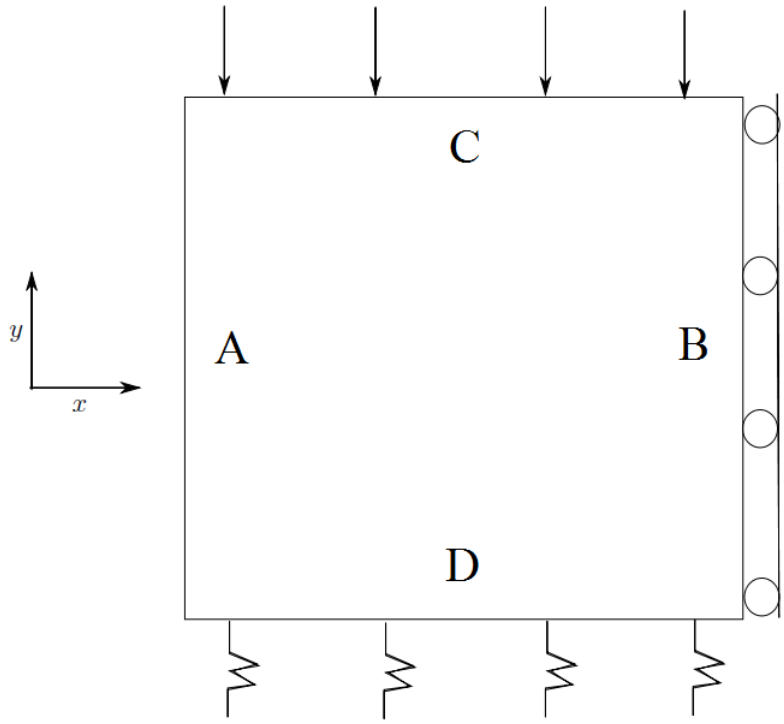


Figure 3.8: Boundary conditions used in FLAC.

Consider the square 2D solid model as illustrated in Fig. (3.8). Boundary A is considered to be a free boundary. This means that grid nodes placed on this boundary can move and deform freely. Boundary B indicates a displacement boundary condition. Nodes that are placed on this boundary are free to move on the surface of the boundary plane, but movement is prohibited normal to this plane. This means that the boundary nodes are not allowed to move over this boundary. Boundary C indicates a stress boundary where a constant external force is applied over the surface of this boundary throughout the simulation. These forces are usually calculated from the initial stress state that is specified at the start of the simulation (usually the virgin stress). Since the stress is applied over an area, Eqs. (2.12) and (2.15) can be used to calculate the external force vector on each of the boundary nodes. The grid nodes that lie on this boundary are free to deform in any direction but they are constantly restricted to this external force vector. Boundary D indicates that the BNS boundary condition is used.

The above mentioned boundary conditions all have respective advantages and disadvantages. Displacement boundary conditions are easy to implement, but since no displacement is allowed over the boundaries, the system might have an *over-stiff* behavior. Stress boundary conditions are also easy to implement and in most cases they give more accurate representations of stress values close to the boundary, but if high concentrations of stress are present in the domain, this type of boundary might *under-respond* and the domain will relax.

Care must be taken in defining and planning the boundaries for a particular problem, as the boundaries, to a large extent, define the solution of the problem. Therefore, for each problem an analysis should be done of the effect of a particular set of boundary conditions, that include the type and position of the boundaries, on the results of the simulation.

3.3 Elemental formulation of the strain tensor

This section assumes a JFLAC domain has been discritised into tetrahedrons as explained in Section (3.1) and that the simulation is in the process of solving for the *equilibrium solution*. The definition of equilibrium solution is explained in Section (3.6). During each time step calculation in the JFLAC algorithm, for each of the four nodes of a tetrahedron, a new position due to certain body forces acting on the domain is obtained. The position for a node at time step t may differ from the position it had at time step $t - \triangle t$. This implies that the tetrahedron deformed from its reference configuration at time step $t-\triangle t$, to its deformed configuration at time step t and due to this deformation, the tetrahedron experiences a strain. This section describes how strain is calculated for a tetrahedron by using the relations in Section (2.1).

Consider a single tetrahedron taken from a stressed JFLAC domain that is not in equilibrium, as illustrated in Fig. (3.9). The tetrahedron nodes are locally labeled N^l where l can take the values 1, 2, 3 or 4. The surface (face) directly opposite N^l is labeled A^l, but

for the purpose of the derivation, expression A^l will be referred to as A^k, where $k = l$.

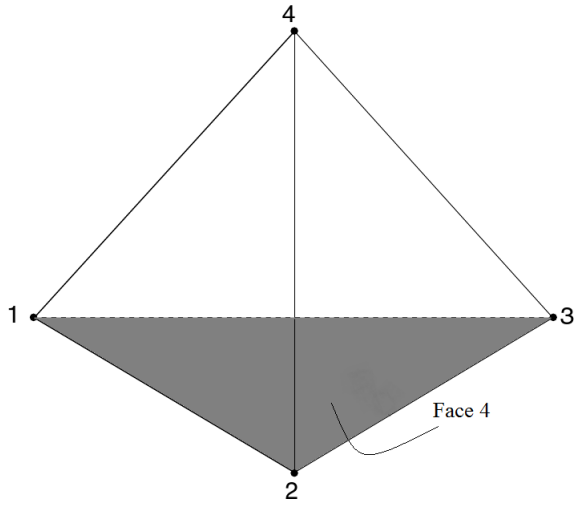


Figure 3.9: Tetrahedron.

Gauss's divergence theorem [1], may be expressed as

$$\int_V (\nabla \cdot \mathbf{v}) dV = \int_S \mathbf{v} \cdot \mathbf{n}\, dS, \tag{3.1}$$

where $\mathbf{v}$ is a continuous vector function, S is a closed surface and $\mathbf{n}$ is the normal vector applicable to a particular surface. Eq. (3.1) can be reformulated to fit the tetrahedron in Fig. (3.9). This yields

$$\int_V v_{i,j} dV = \int_S v_i^{A^k} n_j^{A^k} dS \tag{3.2}$$

where the integrals are taken over the volume and the surfaces of the tetrahedron. A *linear velocity field* is generated in the tetrahedron, since each of the four nodes of the tetrahedron in Fig. (3.9) contains its own velocity. This implies that the tetrahedron contains a *constant strain-rate*. The normal vector over each face of the tetrahedron, $\mathbf{n}^{A^k}$, will also be constant, because of the planar nature of triangles. Hence, Eq. (3.2) becomes:

$$Vv_{i,j} = \sum_{k=1}^{4} \bar{v_i}^{A^k} n_j^{A^k} S^{A^k} \tag{3.3}$$

after integration. Here $\bar{v}_i^{A^k}$ is the average velocity of face A^k. For a linear velocity variation, $\bar{v}_i^{A^k}$ is equal to

$$\bar{v}_i^{A^k} = \frac{1}{3} \sum_{l=1, l\neq k}^{4} v_i^{N^l}, \tag{3.4}$$

where $v_i^{N^l}$ is the velocity of node N^l. Substitution of Eq. (3.4) into Eq. (3.3), and by reorganizing the terms by node contributions, yields

$$Vv_{i,j} = \sum_{k=1}^{4} \left[\left[\frac{1}{3} \sum_{l=1, l\neq k}^{4} v_i^{N^l} \right] n_j^{A^k} S^{A^k} \right]. \tag{3.5}$$

Reorganizing the terms in Eq. (3.5) gives

$$Vv_{i,j} = \frac{1}{3} \sum_{l=1}^{4} v_i^{N^l} \left[\sum_{k=1, k\neq l}^{4} n_j^{A^k} S^{A^k} \right]. \tag{3.6}$$

If v_i in Eq. (3.2) is replaced with 1, it becomes:

$$\sum_{k=1}^{4} n_j^{A^k} S^{A^k} = 0. \tag{3.7}$$

Substituting Eq. (3.7) into Eq. (3.6), dividing by V and replacing expressions $n_j^{A^k}$ and S^{A^k} with expressions $n_j^{A^l}$ and S^{A^l} respectively (since $k = l$), yields

$$v_{i,j} = -\frac{1}{3V} \sum_{l=1}^{4} v_i^{N^l} n_j^{A^l} S^{A^l} \tag{3.8}$$

By substituting the relation for the velocity in Eq. (3.8) into Eq. (2.10) the Euler strain-rate tensor becomes

$$\dot{\epsilon}_{ij} = -\frac{1}{6V}\sum_{l=1}^{4}\left(v_i^{N^l} n_j^{A^l} + v_j^{N^l} n_i^{A^l}\right) S^{A^l}. \tag{3.9}$$

Using Eq. (3.8) together with Eq. (2.11) yields

$$\dot{\omega}_{ij} = -\frac{1}{6V}\sum_{l=1}^{4}(v_i^{N^l} n_j^{A^l} - v_j^{N^l} n_i^{A^l}) S^{A^l} \tag{3.10}$$

for the rate of rotation tensor. For small displacements and displacement gradients during $\triangle t$, the strain increment can be written as

$$\triangle\epsilon_{ij} = \dot{\epsilon}_{ij}\triangle t. \tag{3.11}$$

3.4 The constitutive laws of JFLAC

Assume that for time t, the strain increment, $\triangle\epsilon_{ij}$, is known from Eq. (3.11) for a particular tetrahedron in the system. The next step is to calculate the stress increment, $\triangle\sigma_{ij}$, in the tetrahedron due to $\triangle\epsilon_{ij}$. Hooke's law in Eq. (2.21) is used to calculate the elastic stress increment for this time step. The stress increment is then added to the total accumulated stress, σ_{ij}^T, that was calculated in previous time steps, for this tetrahedron. From this, principal stresses can be calculated, and if the principal stresses are lower than limits set by the Mohr-Coulomb constitutive law, i.e. if $F(\sigma_1, \sigma_3)$ in Eq. (2.34) is less than zero (stresses fall within the yield surface of Fig. (2.7)), then the tetrahedron is considered to be in the elastic region. However, if the calculated principal stresses exceed the Mohr-Coulomb limits, then a correction is made to the total stress by applying a plastic flow rule. This returns the total stress back to the yield surface in Fig. (2.7), and the tetrahedron has failed. It is important to note that when a tetrahedron fails, it is not removed from the system. The stress inside this tetrahedron has simply decreased. It is possible for this tetrahedron to again accumulate stress and fail more than once.

The Mohr-Coulomb constitutive model was implemented in JFLAC since it is widely used in rock mechanics. This model implements a shear yield function according to the

Mohr-Coulomb criterion with a non-associated shear flow rule. This means that a tetrahedron that fails under shear, is permanently deformed and it cannot return to its original state. A tensile yield function (tension cutoff) with an associated flow rule is introduced to the Mohr-Coulomb model to also allow for a tetrahedron to fail under tension. A detailed derivation of the tetrahedron stress calculation for time step t is given.

Assume that the strain increment $\triangle\epsilon_{ij}$ at a point for a tetrahedron taken from the JFLAC system is calculated for time step t through Eq. (3.11). A number of relations must hold such that a plastic flow rule can be applied to the tetrahedron. These relations are:

1. The strain increment can then be decomposed into the sum of the elastic strain increments, $\triangle\epsilon^e_{ij}$, and plastic strain increments, $\triangle\epsilon^p_{ij}$, such that

$$\triangle\epsilon_{ij} = \triangle\epsilon^e_{ij} + \triangle\epsilon^p_{ij}. \tag{3.12}$$

2. A linear relation exists between the stress increment, $\triangle\sigma_{ij}$ and elastic strain increment, $\triangle\epsilon^e_{ij}$ expressed as

$$\triangle\sigma_{ij} = E(\triangle\epsilon^e_{ij}), \tag{3.13}$$

   where E is a function of the elastic stress increment.

3. The plastic strain increments in Eq. (3.12) are given by

$$\triangle\epsilon^p_{ij} = \eta\frac{\partial g}{\partial\sigma_{ij}}, \tag{3.14}$$

   where η is a constant that may depend on space coordinates and g is a function that describes a particular flow rule.

4. Lastly, the newly calculated stress should also satisfy the yield function Eq. (2.34) such that

$$F(\sigma^T_{ij} + \triangle\sigma_{ij}) = 0, \tag{3.15}$$

   and since F in Eq. (3.15) is a linear function of the components of σ_{ij}, it can be expressed as

$$F(\sigma^T_{ij}) + F(\triangle\sigma_{ij}) = 0. \tag{3.16}$$

Once the above relations are set, a new expression for the stress at time step t can be derived. Start by substituting Eq. (3.12) into Eq. (3.13), yielding

$$\triangle\sigma_{ij} = E(\triangle\epsilon_{ij}) - E(\triangle\epsilon^p_{ij}). \tag{3.17}$$

Substitution of Eq. (3.14) into Eq. (3.17) gives

$$\triangle\sigma_{ij} = E(\triangle\epsilon_{ij}) - \eta E\left(\frac{\partial g}{\partial\sigma_{ij}}\right). \tag{3.18}$$

Define the new stress component, σ^N_{ij}, and the elastic guess for the stress, σ^I_{ij} , as

$$\sigma^N_{ij} = \sigma^T_{ij} + \triangle\sigma_{ij}, \tag{3.19}$$
$$\sigma^I_{ij} = \sigma^T_{ij} + E(\triangle\epsilon_{ij}). \tag{3.20}$$

Using the expression of the stress increment in Eq. (3.18) and by definition of the elastic guess, the new stress in Eq. (3.19) becomes:

$$\sigma^N_{ij} = \sigma^I_{ij} - \eta E\left(\frac{\partial g}{\partial\sigma_{ij}}\right) \tag{3.21}$$

after elimination of σ^T_{ij} in Eqs. (3.19) and (3.20). An expression for η will now be derived.

Eq. (3.16) becomes after substitution of Eq. (3.18)

$$F(\sigma^T_{ij}) + F\left[E(\triangle\epsilon_{ij}) - \eta E\left(\frac{\partial g}{\partial\sigma_{ij}}\right)\right] = 0 \tag{3.22}$$

and since F is a linear function of the stress components, Eq. (3.22) can be expressed as

$$F(\sigma^T_{ij}) + F\left[E(\triangle\epsilon_{ij})\right] - F\left[\eta E\left(\frac{\partial g}{\partial\sigma_{ij}}\right)\right] = 0. \tag{3.23}$$

Since F is a homogeneous function, the following holds [19]:

$$F(\beta x) = \beta F(x). \tag{3.24}$$

Using this relation, Eq. (3.23) becomes:

$$F(\sigma_{ij}^T) + F\left[E(\triangle\epsilon_{ij})\right] - \eta F\left[E\left(\frac{\partial g}{\partial \sigma_{ij}}\right)\right] = 0. \tag{3.25}$$

From Eq. (3.20), the first and second terms in Eq. (3.25) can be replaced with $F(\sigma_{ij}^I)$, and consequently an expression for η can then be expressed as

$$\eta = \frac{F(\sigma_{ij}^I)}{F\left[E\left(\frac{\partial g}{\partial \sigma_{ij}}\right)\right]}. \tag{3.26}$$

The above derived relation holds for any constitutive model applied in FLAC. The Mohr-Coulomb condition, as explained in Section (2.2), will be used to describe the relations discussed above in full and will define functions for the flow rules, g, used above in terms of a shear flow function, g^s, and a tensile flow function, g^t.

Again consider a single tetrahedron element taken from the JFLAC domain for time step t. The elastic guess for the stress (σ_{ij}^I) as described by Eq. (3.20), is calculated and added to the total stress, σ_{ij}^T, for that element. Principal stresses are then calculated and sorted such that $\sigma_1 \leq \sigma_2 \leq \sigma_3$. Note that σ_1 is always the largest principal stress, but the $\leq$ signs are used because of the compressive negative convention followed in FLAC.

Assume that the stress has been rotated into its principal coordinate system such that

$$\sigma_{ij}^T = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix}. \tag{3.27}$$

By expanding Fig. (2.5) for a compressive negative sign convention, Fig. (3.10) is generated.

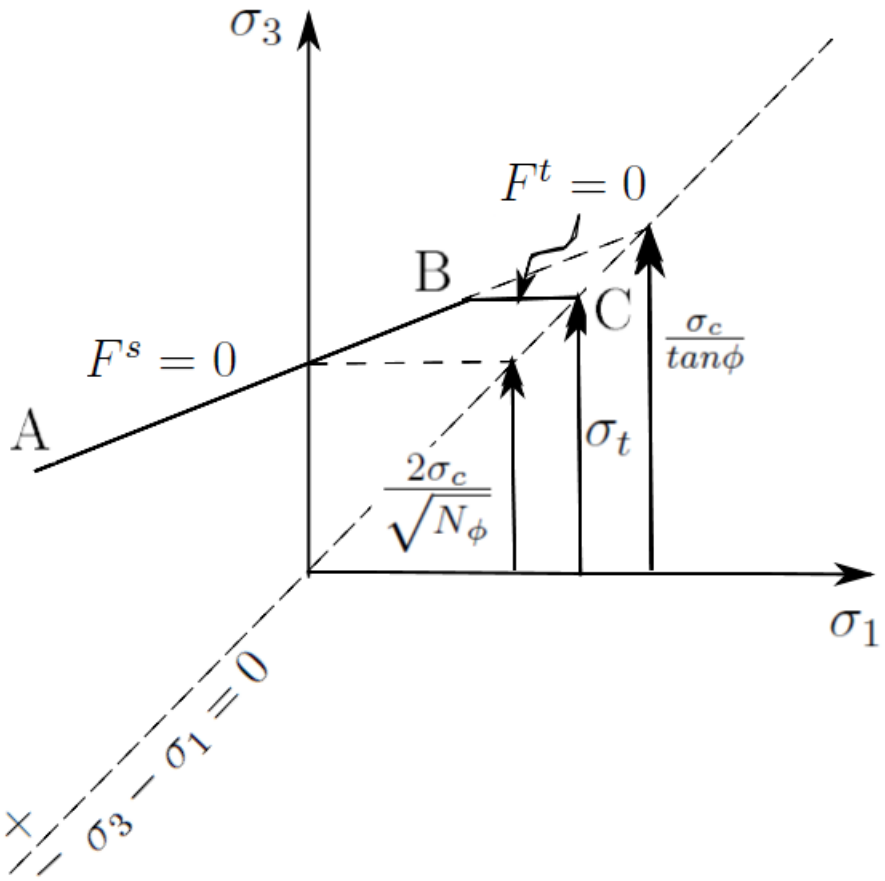


Figure 3.10: Mohr-Coulomb failure envelope with compressive negative stress state.

Fig. (3.10) shows the Mohr-Coulomb criterion with a tension cut-off, represented in $\sigma_1 - \sigma_3$ space for a compressive negative stress state. The failure envelope, $F(\sigma_1, \sigma_3) = 0$ is defined within the interval A - B ($F^s = 0$), following Mohr-Coulomb type behavior, whereas the curve in section B - C ($F^t = 0$) is characterized by a tensile failure condition in which $\sigma_3 \leq \frac{\sigma_c}{\tan\phi}$. Note that the maximum tensile strength of a material is given by [6]

$$\sigma_{t_{max}} = \frac{\sigma_c}{\tan\phi}. \tag{3.28}$$

The yield function is violated where values for σ_1 and σ_3 are found such that $F(\sigma_1, \sigma_3) > 0$, i.e. lying above the line where $\sigma_3 - \sigma_1 = 0$, or if σ_t is exceeded.

The function for F^s can be determined by referring to Eq. (2.31) in Section (2.2) where the linear relation is given as

$$\sigma_1 = \sigma_c + N_\phi \sigma_3, \tag{3.29}$$

with $\sigma_c = -2C_0\sqrt{N_\phi}$ (Eq. (2.33)) for the compressive negative stress state. The function, F^s, then becomes

$$F^s = \sigma_1 - \sigma_3 N_\phi + 2C_0\sqrt{N_\phi}, \tag{3.30}$$

and corresponds to the Mohr-Coulomb failure criterion. For the tension cut-off, point B to point C in Fig. (3.10), a tension failure criterion is derived from Fig. (3.10), and is given as

$$F^t = \sigma_3 - \sigma_t. \tag{3.31}$$

Shear plastic flow and tensile plastic flow are described by two functions, g^s and g^t, respectively. The function g^s corresponds to a non-associated law and has the form

$$g^s = \sigma_1 - \sigma_3 N_\psi, \tag{3.32}$$

where ψ is the dilatency angle (angle of deformation). N_ψ is given by

$$N_\psi = \frac{1 + \sin\psi}{1 - \sin\psi}. \tag{3.33}$$

The function g^t corresponds to an associated flow rule and is written as

$$g^t = -\sigma_3, \tag{3.34}$$

providing a relation for the magnitude of the plastic strain increment vector [6].

The plastic correction that applies to the shear flow rule can be found by differentia-

ting Eq. (3.32), [6] yielding

$$\frac{\partial g^s}{\partial \sigma_1} = 1, \tag{3.35}$$

$$\frac{\partial g^s}{\partial \sigma_2} = 0, \tag{3.36}$$

and

$$\frac{\partial g^s}{\partial \sigma_3} = -N_\psi. \tag{3.37}$$

The incremental form of Hooke's law, expressed in terms of stress increments, strain increments and material constants α_1 and α_2, has the form

$$\triangle\sigma_1 = \alpha_1 \triangle\epsilon_1^e + \alpha_2(\triangle\epsilon_2^e + \triangle\epsilon_3^e), \tag{3.38}$$

$$\triangle\sigma_2 = \alpha_1 \triangle\epsilon_2^e + \alpha_2(\triangle\epsilon_1^e + \triangle\epsilon_3^e), \tag{3.39}$$

$$\triangle\sigma_3 = \alpha_1 \triangle\epsilon_3^e + \alpha_2(\triangle\epsilon_1^e + \triangle\epsilon_2^e). \tag{3.40}$$

where

$$\alpha_1 = K + \frac{4}{3}G \tag{3.41}$$

and

$$\alpha_2 = K - \frac{2}{3}G \tag{3.42}$$

is Young's modulus and Poisson ratio respectively [5]. An expression for Eq. (3.13) in terms of the principal stresses may be expressed as

$$\triangle\sigma_i = E_i(\triangle\epsilon_j^e) \qquad i, j = 1, 2, 3. \tag{3.43}$$

Note that Eq. (3.43) is not expressed in tensor form, but rather in terms of the principal components. Rewriting Eqs. (3.38), (3.39) and (3.40) in the form of Eq. (3.43), gives

$$E_1(\triangle\epsilon_1^e, \triangle\epsilon_2^e, \triangle\epsilon_3^e) = \alpha_1 \triangle\epsilon_1^e + \alpha_2(\triangle\epsilon_2^e + \triangle\epsilon_3^e), \tag{3.44}$$

$$E_2(\triangle\epsilon_1^e, \triangle\epsilon_2^e, \triangle\epsilon_3^e) = \alpha_1 \triangle\epsilon_2^e + \alpha_2(\triangle\epsilon_1^e + \triangle\epsilon_3^e), \tag{3.45}$$

$$E_3(\triangle\epsilon_1^e, \triangle\epsilon_2^e, \triangle\epsilon_3^e) = \alpha_1 \triangle\epsilon_3^e + \alpha_2(\triangle\epsilon_1^e + \triangle\epsilon_2^e). \tag{3.46}$$

Replacing expressions $\triangle\epsilon_1^e$, $\triangle\epsilon_2^e$ and $\triangle\epsilon_3^e$ in Eqs. (3.44), (3.45) and (3.46) with $\frac{\partial g^s}{\partial \sigma_1}$, $\frac{\partial g^s}{\partial \sigma_2}$ and $\frac{\partial g^s}{\partial \sigma_3}$ (from Eq. (3.14)) respectively, yields

$$E_1(\frac{\partial g^s}{\partial \sigma_1}, \frac{\partial g^s}{\partial \sigma_2}, \frac{\partial g^s}{\partial \sigma_3}) = \alpha_1 - \alpha_2 N_\psi, \tag{3.47}$$

$$E_2(\frac{\partial g^s}{\partial \sigma_1}, \frac{\partial g^s}{\partial \sigma_2}, \frac{\partial g^s}{\partial \sigma_3}) = \alpha_2(1 - N_\psi), \tag{3.48}$$

$$E_3(\frac{\partial g^s}{\partial \sigma_1}, \frac{\partial g^s}{\partial \sigma_2}, \frac{\partial g^s}{\partial \sigma_3}) = -\alpha_1 N_\psi + \alpha_2. \tag{3.49}$$

Expressions for the new stress can be found by substituting Eqs. (3.47), (3.48) and (3.49) into Eq. (3.21), yielding

$$\sigma_1^N = \sigma_1^I - \eta^s(\alpha_1 - \alpha_2 N_\psi), \tag{3.50}$$

$$\sigma_2^N = \sigma_2^I - \eta^s \alpha_2(1 - N_\psi), \tag{3.51}$$

$$\sigma_3^N = \sigma_3^I - \eta^s(-\alpha_1 N_\psi + \alpha_2). \tag{3.52}$$

By replacing the expression F with F^s in Eq. (3.30), the denominator of Eq. (3.26) becomes:

$$F^s\left[E(\frac{\partial g}{\partial \sigma_{ij}})\right] = (\alpha_1 - \alpha_2 N_\psi) - N_\phi(-\alpha_1 N_\psi + \alpha_2). \tag{3.53}$$

Substitution of Eqs. (3.30) and (3.53) in Eq. (3.26) then gives

$$\eta^s = \frac{\sigma_1^I - \sigma_3^I N_\phi + 2C_0\sqrt{N_\phi},}{(\alpha_1 - \alpha_2 N_\psi) - N_\phi(-\alpha_1 N_\psi + \alpha_2)}. \tag{3.54}$$

Tensile failure is derived in a similar fashion, and by differentiating Eq. (3.34), it follows that

$$\frac{\partial g^t}{\partial \sigma_1} = 0, \tag{3.55}$$

$$\frac{\partial g^t}{\partial \sigma_2} = 0, \tag{3.56}$$

$$\frac{\partial g^t}{\partial \sigma_3} = -1. \tag{3.57}$$

Replacing expressions $\triangle\epsilon_1^e$, $\triangle\epsilon_2^e$ and $\triangle\epsilon_3^e$ in Eqs. (3.44), (3.45) and (3.46) with $\frac{\partial g^t}{\partial \sigma_1}$, $\frac{\partial g^t}{\partial \sigma_2}$ and $\frac{\partial g^t}{\partial \sigma_3}$ (from Eq. (3.14)) respectively, yields

$$E_1(\frac{\partial g^t}{\partial \sigma_1}, \frac{\partial g^t}{\partial \sigma_2}, \frac{\partial g^t}{\partial \sigma_3}) = -\alpha_2, \tag{3.58}$$

$$E_2(\frac{\partial g^t}{\partial \sigma_1}, \frac{\partial g^t}{\partial \sigma_2}, \frac{\partial g^t}{\partial \sigma_3}) = -\alpha_{2,} \tag{3.59}$$

$$E_3(\frac{\partial g^t}{\partial \sigma_1}, \frac{\partial g^t}{\partial \sigma_2}, \frac{\partial g^t}{\partial \sigma_3}) = -\alpha_1. \tag{3.60}$$

Expressions for the new stress can be found by substituting Eqs. (3.58), (3.59) and (3.60) into Eq. (3.21), yielding

$$\sigma_1^N = \sigma_1^I + \eta^t \alpha_{2,} \tag{3.61}$$

$$\sigma_2^N = \sigma_2^I + \eta^t \alpha_{2,} \tag{3.62}$$

$$\sigma_3^N = \sigma_3^I + \eta^t \alpha_1. \tag{3.63}$$

By replacing the expression F with F^t in Eq. (3.31), the denominator of Eq. (3.26) becomes:

$$F^t \left[E(\frac{\partial g}{\partial \sigma_{ij}}) \right] = \alpha_1. \tag{3.64}$$

Lastly, substitution of Eqs. (3.31) and (3.64) into Eq. (3.26), gives

$$\eta^t = \frac{\sigma_3^I - \sigma_t}{\alpha_1}. \tag{3.65}$$

3.5 Mixed Discritization applied on strain-rate and stress

Although tetrahedrons have the advantage of not forming hourglass deformations, a tetrahedron still does not have enough modes of deformation. Tetrahedrons cannot deform individually without a change in volume as required by constitutive laws. This is called volumetric locking. Numerical anomalies can occur in areas where high gradients of stresses and deformations are expected. This often happens in the fully plastic range.

Nagtegaal et al. [15] states that it is often found that materials in the plastic range exhibit an over-stiff tangent behavior. The materials often exceed the limit load and in some cases they contain no load at all. It was also shown that the cause of this inaccuracy is the incremental deformation fields of three-dimensional elements that are highly constrained at or near the limit load. The difficulty arises because, under the in-compressibility condition, certain classes of meshes are over-constrained. One way to resolve this issue is to increase the order of the element. However, a drawback of introducing additional degrees of freedom can cause hour-glassing to occur, as described in Section (3.1).

Marti and Cundall [12] have proposed a procedure that reduces the probability of obtaining unwanted hourglass deformations in the system. The technique is called Mixed Discritization (MD), and is applied to the strain- and stress rate of the 5 tetrahedrons associated with each zone in the system. This is described below.

3.5.1 Mixed Discritization (MD) applied to the strain-rate

Consider a hexahedral zone corresponding to an assembly of 5 tetrahedrons. Once the strain-rate tensor of Eq. (3.9) is calculated for all the tetrahedrons in the system, MD can be applied to the tetrahedrons associated with that zone. The strain-rates for each tetrahedron are decomposed into the volumetric and deviatoric components by applying a simple tensor analysis technique, i.e.

$$\dot{\epsilon}_{ij}^{T^h} = \dot{\zeta}_{ij}^{T^h} + \frac{\dot{\epsilon}^{T^h}}{3}\delta_{ij} \tag{3.66}$$

where T^h represents one tetrahedron of the hexahedral zone, with $h = 1, ..., 5$. In Eq. (3.66) $\dot{\zeta}_{ij}^{T^h}$ is the deviatoric part of the strain-rate tensor for tetrahedron, T^h, and $\dot{\epsilon}^{T^h}$ is the first strain-rate invariant of the tetrahedron, expressed as

$$\dot{\epsilon}^{T^h} = \dot{\epsilon}_{ii}^{T^h}. \tag{3.67}$$

The first invariant for the zone is then calculated as the volumetric strain-rate average over all the tetrahedrons in the zone

$$\dot{\bar{\epsilon}} = \frac{\sum_{h=1}^{5} \dot{\epsilon}^{T^h} V^{T^h}}{\sum_{h=1}^{5} V^{T^h}}, \tag{3.68}$$

where V^{T^h} is the volume of tetrahedron T^h, and $\dot{\bar{\epsilon}}$ is the volumetric strain average of all the tetrahedrons in the zone. Finally, the volumetric strain-rate in Eq. (3.66) is replaced by the average zone volumetric strain-rate $\bar{\epsilon}$ in Eq. (3.68). This is expressed as

$$\dot{\epsilon}_{ij}^{T^h} = \dot{\zeta}_{ij}^{T^h} + \frac{\dot{\bar{\epsilon}}}{3}\delta_{ij}. \tag{3.69}$$

3.5.2 Mixed Discritization on stress

The application of MD to the stress inside a hexahedral zone is similar to the technique described in Section (3.5.1). The stress estimate for each tetrahedron inside the zone is decomposed into the deviatoric and volumetric parts by

$$\sigma_{ij}^{T^h} = \chi_{ij}^{T^h} + \frac{\sigma^{T^h}}{3}\delta_{ij} \tag{3.70}$$

where $\chi_{ij}^{T^h}$ is known as the deviatoric part of the stress tensor and $\sigma^{T^h} = \sigma_{ii}^{T^h}$. The volumetric average for all the tetrahedrons in the zone is calculated from

$$\overline{\sigma} = \frac{\sum_{h=1}^{5} \sigma^{T^h} V^{T^h}}{\sum_{h=1}^{5} V^{T^h}}. \tag{3.71}$$

Finally the tetrahedron stress tensors for all the tetrahedrons in the zone are calculated by substituting the volumetric average calculated in Eq. (3.71) into Eq. (3.70), yielding

$$\sigma_{ij}^{T^h} = \chi_{ij}^{T^h} + \frac{\overline{\sigma}}{3}\delta_{ij}. \tag{3.72}$$

3.5.3 Mixed Nodal Discritization applied to the strain-rate

When a tetrahedron-only model is used as input in FLAC, MD cannot be applied to the tetrahedrons that share the same hexahedral zone since it may not be possible to identify any hexahedral zones. Consequently the Mixed Nodal Discritization (MND) procedure was developed that operates in a very similar fashion to MD. The basic calculation sequence is kept the same as when applying MD; however, an averaging process for the strain-rates and stresses is performed for tetrahedrons that share the same node and not the same zone. MND can also be applied to a model that contains hexahedron zones. This procedure is explained below.

The strain-rate tensor, $\dot{\epsilon}_{ij}$, is calculated from Eq. (3.9) for a particular time step, t, and is decomposed into volumetric and deviatoric parts:

$$\dot{\epsilon}_{ij}^{T^h} = \dot{\zeta}_{ij}^{T^h} + \frac{\dot{\epsilon}^{T^h}}{3}\delta_{ij}. \tag{3.73}$$

The volumetric parts of all the tetrahedrons connected to a global node, N^z from in the system, are calculated by

$$\dot{\bar{\epsilon}}^{N^z} = \frac{\sum_{h=1}^{w} \dot{\epsilon}^{T^h} V^{T^h}}{\sum_{h=1}^{w} V^{T^h}}, \tag{3.74}$$

where w is the number of tetrahedrons that share the same node. Eq. (3.74) is known as the nodal volumetric strain-rate. This is similar to the MD technique described in the previous section. After nodal volumetric strain-rate values are obtained, a mean value for the volumetric strain-rate is calculated by taking the average of the volumetric strain-rate values of its four nodes. Mathematically this is expressed as

$$\dot{\bar{\epsilon}}^{T^h} = \frac{1}{d}\sum_{l=1}^{d} \dot{\bar{\epsilon}}^{N^l}. \tag{3.75}$$

Finally, the tetrahedron strain-rate is redefined by superimposing the deviatoric part and volumetric average in Eq. (3.75) to give

$$\epsilon_{ij} = \dot{\eta}_{ij} + \dot{\bar{\epsilon}}\delta_{ij}. \tag{3.76}$$

3.5.4 Nodal Mixed Discritization on stress

Consider Hooke's law expressed in the form [10],

$$\triangle\sigma = E(\dot{\overline{\epsilon}}^{T^h} - \triangle\epsilon^{p^{T^h}}), \tag{3.77}$$

where $\triangle\sigma$ is the stress increment and $\triangle\epsilon^{p^{T^h}}$ is the plastic volumetric strain increment. MND will be applied to the term $E\triangle\epsilon^{p^{T^h}}$. Let $E\triangle\epsilon^{p^{T^h}}$ be known as the plastic stress increment $\triangle\sigma^{p^{T^h}}$ of the tetrahedron. Eq. (3.77) can now be expressed as

$$\triangle\sigma = E\dot{\overline{\epsilon}}^{T^h} - \triangle\sigma^{p^{T^h}}. \tag{3.78}$$

The MND that is applied to $\triangle\sigma^{p^{T^h}}$ is similar to the MND applied on strain in Section (3.5.3). Nodal values for $\triangle\sigma^{p^{N^z}}$ are calculated as the weighted average of all the tetrahedrons that share node, N^z. Mathematically this is expressed as

$$\triangle\sigma^{p^{N^z}} = \frac{\sum_{h=1}^{w} \triangle\sigma^{p^{T^h}} V^{T^h}}{\sum_{h=1}^{w} V^{T^h}}. \tag{3.79}$$

Again, similar to strain MND, after the nodal values, $\triangle\sigma^{p^{N^z}}$, are obtained, a mean value for the plastic stress inside a tetrahedron is calculated by taking the average of its four nodal values by

$$\triangle\overline{\sigma}^{p^{T^h}} = \frac{1}{d}\sum_{l=1}^{d} \triangle\sigma^{p^{N^l}}. \tag{3.80}$$

Lastly, the stress at each tetrahedron is corrected by substituting the average $\triangle\overline{\sigma}^{p^{T^h}}$ for $\triangle\sigma^{p^{T^h}}$ in Eq. (3.78). This yields

$$\sigma_{ij}^{n^{T^h}} = \sigma_{ij}^{o^{T^h}} + [E\dot{\overline{\epsilon}}^{T^h} - \triangle\overline{\sigma}^{p^{T^h}}]\delta_{ij} \tag{3.81}$$

where $\sigma_{ij}^{n^{T^h}}$ is the new stress of the tetrahedron and $\sigma_{ij}^{o^{T^h}}$ is the old stress of the tetrahedron before MND is applied.

3.6 Nodal formulation of the equilibrium equations

The equilibrium equations described in Section (2.3) must be applied to all the nodes of the Lagrangian grid for each time step in the JFLAC simulation. For each time step calculation in the algorithm, body forces of the tetrahedral elements are mapped to their respective nodes. Once finished, the *out-of-balance* force at every node is analyzed. The node with the largest out-of-balance force is used to determine whether the system is in a *equilibrium* state. If the out-of-balance force is larger than a user specified threshold, the sequence will repeat for another time step. However, if this force is below the given threshold, the system is assumed to be in equilibrium. This section describes how the out-of-balance force for a particular node in the system is calculated.

The principle of *virtual work* [3] is a convenient way to treat the laws of motion. From Gauss's divergence theorem, as given in Eq. (3.1), it follows that the work performed on a generic surface, is equivalent to the flux of energy transferred from the environment into the system, increasing its energy. However, a tetrahedron would rather absorb this energy by deforming and releasing the energy internally by moving the nodes of the tetrahedron. From Eq. (3.1) it follows that

$$W^{int} = \int_V (\nabla \cdot \mathbf{v}) dV \tag{3.82}$$

and

$$W^{ext} = \int_{\mathbf{S}} \mathbf{v} \cdot \mathbf{n}\, dS \tag{3.83}$$

where W^{ext} denotes the *external work* applied to the system and W^{int} denotes the *internal work* of the system. It also follows that

$$W^{ext} = W^{int}. \tag{3.84}$$

The difference approximation adopted in JFLAC assumes that the domain consists of an assembly of *constant-strain* tetrahedrons that is subjected to body forces b_i. The theorem of virtual work will be used to derive the nodal forces, $F_i^{N^l}$, with $l = 1, .., 4$, that acts on a single tetrahedron, T^z, inside the global system that is in static equilibrium, from the

tetrahedron stresses and body forces. If a virtual velocity, $\delta v_i^{N^l}$, is applied over each of the four tetrahedron nodes, it will generate a *linear velocity field,* δv_i, inside the tetrahedron with a *constant strain-rate*, $\delta\dot{\epsilon}_{ij}$. Consider the Cauchy equation of Eq.(2.44) given by

$$\sigma_{ij,j} + \rho b_i = \rho \frac{dv_i}{dt}. \tag{3.85}$$

Eq. (3.85) can also be expressed as

$$\sigma_{ij,j} + B_i = 0 \tag{3.86}$$

where

$$B_i = \rho(b_i - \frac{dv_i}{dt}) \tag{3.87}$$

For the tetrahedron mentioned above, the *external work-rate,* w^{ext}, is due to the forces acting on the four nodes of the tetrahedron, $F_i^{N^l}$, and the total body force, B_i of the tetrahedron. The *internal work-rate,* w^{int}, is done by the stresses, σ_{ij}, on the surfaces of the tetrahedron. From Eq. (3.84) it follows that the external- and internal work rates should be the same under $\delta v_i^{N^l}$. The external- and internal work rates can then be expressed as

$$w^{ext} = \sum_{l=1}^{4} \delta v_i^{N^l} F_i^{N^l} + \int_V \delta v_i B_i dV \tag{3.88}$$

and

$$w^{int} = \int_V \delta\dot{\epsilon}_{ij}\sigma_{ij} dV. \tag{3.89}$$

By substitution of Eq. (3.9) for a *constant strain-rate* tetrahedron, Eq. (3.89) becomes

$$w^{int} = -\frac{1}{6}\sum_{l=1}^{4}\left(\delta v_i^{N^l}\sigma_{ij}n_j^{A^l} + \delta v_j^{N^l}\sigma_{ij}n_i^{A^l}\right)S^{A^l}. \tag{3.90}$$

Since the stress tensor is symmetric, Eq. (3.90) can be simplified to

$$w^{int} = -\frac{1}{3}\sum_{l=1}^{4}\delta v_i^{N^l}\sigma_{ij}n_j^{A^l}S^{A^l}. \tag{3.91}$$

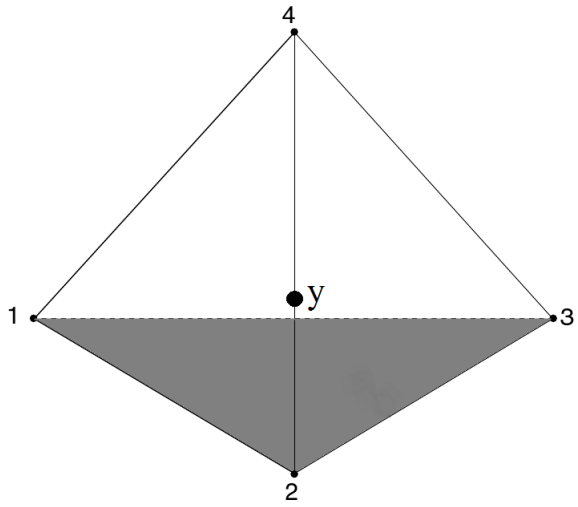


Figure 3.11: Local coordinate system at the tetrahedron centroid.

Also, substituting Eq. (3.87) into Eq. (3.88) gives

$$w^{ext} = \sum_{l=1}^{4} \delta v_i^{N^l} F_i^{N^l} + \int_V \delta v_i \rho b_i dV - \int_V \delta v_i \rho \frac{dv_i}{dt} dV \tag{3.92}$$

and by letting the second and third terms on the right hand side of Eq. (3.92) be equal to

$$w^{ext^b} = \int_V \delta v_i \rho b_i dV \tag{3.93}$$

and

$$w^{int^I} = -\int_V \delta v_i \rho \frac{dv_i}{dt} dV \tag{3.94}$$

where w^{ext^b} is the external work rate contribution of body forces and w^{int^I} is the external work rate contribution of internal forces, Eq. (3.92) becomes

$$w^{ext} = \sum_{l=1}^{4} \delta v_i^{N^l} F_i^{N^l} + w^{ext^b} + w^{int^I}. \tag{3.95}$$

Consider the centroid, or center of mass, $\mathbf{y}$, of the tetrahedron as shown in Fig. (3.11). The centroid of the tetrahedron can be determined by taking the average of the positions

of the four nodes' coordinates. Mathematically the position of the centroid is expressed as

$$\mathbf{y} = \frac{1}{4}\sum_{l=1}^{4} \mathbf{r}^{N^l} \tag{3.96}$$

where $\mathbf{r}^{N^l}$ represents the position vector of node N^l. An expression for the velocity of the tetrahedron at $\mathbf{y}$ can also be expressed as

$$\delta v_i^{\mathbf{y}} = \frac{1}{4}\sum_{l=1}^{4} \delta v_i^{N^l}. \tag{3.97}$$

By analyzing Eq. (3.93) in terms of the velocity at the centroid of the tetrahedron (Eq. (3.97)), it becomes

$$w^{ext^b} = \int_V \frac{1}{4}\sum_{l=1}^{4} \delta v_i^{N^l} \rho b_i dV. \tag{3.98}$$

If the body force in Eq. (3.98) is constant, then it can be simplified to

$$w^{ext^b} = \frac{\rho b_i V}{4}\sum_{l=1}^{4} \delta v_i^{N^l}. \tag{3.99}$$

Substitution of Eq. (3.97) into Eq. (3.94) also gives

$$w^{int^I} = -\int_V \frac{1}{4}\sum_{l=1}^{4} \delta v_i^{N^l} \rho \frac{dv_i}{dt} dV. \tag{3.100}$$

Since ρ is constant inside the tetrahedron, and for small variations in the acceleration field around the centroid of the tetrahedron, Eq. (3.100) can be reduced to

$$w^{int^I} = -\frac{\rho V}{4}\sum_{l=1}^{4} \delta v_i^{N^l} a_i^{N^l} \tag{3.101}$$

where $a_i^{N^l} = \left(\frac{dv_i}{dt}\right)^{N^l}$. Combining Eqs. (3.100) and (3.101) with Eq. (3.92) yields

$$w^{ext} = \sum_{l=1}^{4} \delta v_i^{N^l} F_i^{N^l} + \frac{\rho b_i V}{4} \sum_{l=1}^{4} \delta v_i^{N^l} - \frac{\rho V}{4} \sum_{l=1}^{4} \delta v_i^{N^l} a_i^{N^l}. \tag{3.102}$$

By simplifying Eq. (3.102), the final solution for the external work-rate becomes:

$$w^{ext} = \sum_{l=1}^{4} \delta v_i^{N^l} \left[F_i^{N^l} + \frac{\rho b_i V}{4} - \frac{\rho V}{4} a_i^{N^l} \right]. \tag{3.103}$$

Since Eq. (3.84) holds for the tetrahedron in static equilibrium, Eq. (3.103) can be equated with Eq. (3.91) to give

$$\sum_{l=1}^{4} \delta v_i^{N^l} \left[F_i^{N^l} + \frac{\rho b_i V}{4} - \frac{\rho V}{4} a_i^{N^l} \right] = -\frac{1}{3} \sum_{l=1}^{4} \delta v_i^{N^l} \sigma_{ij} n_j^{A^l} S^{A^l}, \tag{3.104}$$

and since the external work-rate is equal to the internal work-rate for *any* virtual velocity, $\delta v_i^{N^l}$, in static equilibrium, it follows from Eq. (3.104) that

$$-F_i^{N^l} = \frac{\sigma_{ij} n_j^{A^l} S^{A^l}}{3} + \frac{\rho b_i V}{4} - \frac{\rho V}{4} a_i^{N^l}. \tag{3.105}$$

Let the mass, $\frac{\rho V}{4}$, in the inertial term of Eq. (3.105) be replaced by a fictitious nodal mass m^{N^l} to become

$$-F_i^{N^l} = \frac{\sigma_{ij} n_j^{A^l} S^{A^l}}{3} + \frac{\rho b_i V}{4} - m^{N^l} a_i^{N^l}. \tag{3.106}$$

For the system to be in static equilibrium, it must hold that Eq. (3.106) is equal to zero for every node of the system. This implies that Eq. (3.106) becomes:

$$\frac{\sigma_{ij} n_j^{A^l} S^{A^l}}{3} + \frac{\rho b_i V}{4} - m^{N^l} a_i^{N^l} = 0 \tag{3.107}$$

Consider Newton's second law for a node taken from the global system (N^z), expressed as

$$F_i^{N^z} = M^{N^z} a_i^{N^z}. \tag{3.108}$$

The mass term on right hand side of Eq. (3.108) may be expressed as

$$M^{N^z} = [m]^z \tag{3.109}$$

where [.] represents the sum of the mass contributions of all the tetrahedrons connected to N^z. The out-of-balance force may also be expressed as

$$F_i^{N^z} = [\frac{\sigma_{ij} n_j^{A^l} S^{A^l}}{3} + \frac{\rho b_i V}{4}]^{N^z} \tag{3.110}$$

when analyzed in terms of contributions of all connecting tetrahedrons.

3.7 Explicit Finite Difference approximations to the time derivatives

In the preceding sections a finite volume approach was followed for the discritisation process. However, for the time dependence of the algorithm, a finite difference approximation (refer to Appendix (A)) will be used to calculate the equations of motion at the grid nodes. It is also useful at this point to mention that explicit finite difference equations will be used to derive the nodal velocities and displacements for a particular time step. Explicit methods calculate the state of a system for a later time step from the state of the system at the current time, whereas implicit methods find the solution by solving an equation involving both the current state of the system and the later one. Explicit methods mathematically expressed are

$$Y(t + \triangle t) = F[Y(t)] \tag{3.111}$$

where $\triangle t$ is the difference between two consecutive time steps, Y is the current state of the system and $Y(t + \triangle t)$ is the state at a later time.

Implicit methods involve solving an equation of the form

$$H[Y(t), Y(t + \triangle t)] = 0. \tag{3.112}$$

It is clear from Eq. (3.112) that implicit methods are more difficult to implement and it requires more computational time to solve. The general use for implicit methods is when a problem exhibit a stiff behavior. If an explicit method is used to solve such a problem, it may require implementing an impractically small time step to calculate a reasonable estimation to the problem. It may therefore take less computational time to rather use an implicit method with larger time steps.

To illustrate the explicit nature of the algorithm, consider the nodal formulation of the equation of motion, Eq. (3.108), expressed as a system of ordinary differential equations:

$$\left(\frac{\triangle v_i}{\triangle t}\right)^{N^z} = \frac{1}{M^{N^z}} F_i^{N^z} \tag{3.113}$$

where $\left(\frac{\triangle v_i}{\triangle t}\right)^{N^z} = a_i^{N^z}$. Eq. (3.113) is explicitly solved by a finite difference formulation in time. The velocity of node, N^z, is assumed to vary linearly over time interval, $\triangle t$, and is evaluated using the *central difference scheme* as in Eq. (A.19). By evaluating the derivative on the left hand side of Eq. (3.113) by half time steps, $\frac{\triangle t}{2}$, with respect to displacements and forces, the central difference scheme yields

$$\left(\frac{\triangle v_i}{\triangle t}\right)^{N^z} = \frac{v_i^{N^z}(t + \frac{\triangle t}{2})}{2\frac{\triangle t}{2}} - \frac{v_i^{N^z}(t - \frac{\triangle t}{2})}{2\frac{\triangle t}{2}} = \frac{1}{M^{N^z}} F_i^{N^z}. \tag{3.114}$$

With further manipulation Eq. (3.114) becomes

$$v_i^{N^z}(t + \frac{\triangle t}{2}) = v_i^{N^z}(t - \frac{\triangle t}{2}) + \frac{1}{M^{N^z}} F_i^{N^z}. \tag{3.115}$$

The nodal displacements are then updated by using [10]

$$u_i^{N^z}(t+\triangle t) = u_i^{N^z}(t) + \triangle t v_i^{N^z}(t+\frac{\triangle t}{2}). \tag{3.116}$$

3.8 Time step determination

The time interval, $\triangle t$, between two consecutive time steps has to be chosen carefully. A time interval that is too small will increase simulation time. A time interval that is too large might cause the system to become unstable, in which case re-meshing may be required. The finite difference approximations obtained in Eqs. (3.115) and (3.116) will not provide valid answers unless the system is stable.

A widely used technique for choosing a time interval for explicit time marching finite differencing applications is known as the Courant–Friedrichs–Lewy (CFL) condition [4] . The CFL condition is a necessary condition for stability while solving hyperbolic partial differential equations numerically. This assumes that if a pressure wave is traveling through a discrete grid, then the time interval must be less than the time required for the wave to travel between two adjacent grid nodes. Mathematically, the CFL condition can be expressed as

$$\triangle t \leq C\frac{\triangle x}{v_{max}} \tag{3.117}$$

where C is called the Courant number, $\triangle x$ the smallest distance between two adjacent grid nodes, and v_{max} the maximum pressure wave velocity in the medium. When

$$\triangle t = C\frac{dx}{v_{max}}, \tag{3.118}$$

it is known as the Courant limit and $\triangle t$ is at its maximum possible value for the specific simulation. The pressure wave velocity of a JFLAC simulation, v_{max}, usually depends on material properties of the tetrahedrons in the domain. The Courant number C may depend on the application. According to Madariaga [11] C is found to be 0.606 for 2D problems and 0.494 for 3D problems. These values for C allow for $\triangle t$ to be as close to the Courant limit as possible. This improves the run-time, and it improves the accuracy of the

simulation. In some cases where a domain undergoes large deformations for a particular time step, it might be necessary to decrease $\triangle t$ for the system to remain in a stable state. One representation for v_{max} in terms of the Lamé constant λ_L, shear modulus G and material density ρ may be given as

$$v_{max} = \sqrt{\frac{\lambda_L + 2G}{\rho}} \tag{3.119}$$

where the relationship between λ_L, Young's modulus and the Poisson ratio can be expressed as 5

$$\lambda_L = \frac{\nu E}{(1+\nu)(1-2\nu)}. \tag{3.120}$$

The CFL condition is adopted in FLAC. However, the authors chose an alternative approach to time step determination. Since the nodal formulations of Newton's second law, given by

$$F^{N^z} = M^{N^z} \frac{dv^{N^z}}{dt}, \tag{3.121}$$

are always solved for the equilibrium solution, $\triangle t$ is chosen fixed as 1. The nodal masses, M^{N^z}, are then scaled accordingly such that the right hand side of Eq. (3.121) still holds. This speeds up the simulation and the system reaches its equilibrium solution more efficiently.

3.9 Nodal motion damping

Damping is any effect that tends to reduce the amplitude of oscillations in an harmonic oscillatory system. In most cases when a JFLAC system is solved for the equilibrium solution, the total simulation time needed to reach this equilibrium state, as well as how the state is reached, are not considered too important. A user is only interested in the final result. For this reason, the equations of motion at the node points can be damped such that the system reaches the equilibrium solution faster. If the system is not damped, one might find that the system oscillates, causing the system to take longer to reach an equilibrium state.

Several damping schemes may be applied at the grid nodes. Viscous damping is a physical form of damping where a damping factor, such as friction, is introduced into the system. This system might loose most of its energy through heat generation and the oscillatory behavior of the system will quickly fade. However, the default damping scheme used in JFLAC is a non-viscous damping scheme. This type of damping is an artificial form of damping where a part of the system velocity is artificially removed from the system.

Assume that for a particular time step t for the system, the out-of-balance force, $F_i^{N^z}$, has been calculated from Eq. (3.110). Non-viscous damping is then applied to $F_i^{N^z}$ by adding a damping force term, $F_i^{d^{N^z}}$. The damping force term is expressed as

$$F_i^{d^{N^z}} = \gamma s_{sign} F_i^{N^z} \tag{3.122}$$

where

$$s_{sign} = \begin{cases} +1, & for\, v_i^{N^z} > 0 \\ -1. & for\, v_i^{N^z} < 0 \\ 0 & for\, v_i^{N^z} = 0 \end{cases} \tag{3.123}$$

and γ is known as the damping constant. A good value for the constant γ found through experimentation is 0.8.

3.10 Small-strain and large-strain mode in JFLAC

Suppose that a JFLAC system is in the process of solving for the equilibrium solution. If the maximum of all the calculated tetrahedron strain rates of Eq. (3.9) is small for any two sequential time steps, it could be assumed that the system is near equilibrium. Small Strain Mode (SSM) is an optimization of the algorithm that can be applied to a system close to equilibrium. Since the strain-rates are small, it implies that the deformations are small and therefore nodal displacement increments are small. In this case, the grid node positions for the Lagrangian grid do not need to be updated.

When the maximum system strain-rate is large, the system is assumed to be in Large

Strain Mode (LLM) and it follows that grid node positions need updating.

3.11 Implementation of JFLAC

Before the calculation sequence of the JFLAC algorithm can be explained, the input parameters of the algorithm must first be defined. JFLAC requires three files to operate, as shown in Fig. (3.12). Examples of the input files can be found in Appendix (B). These files are generated using custom Java interfaces and do not form part of the JFLAC program. The files have to first be created before the JFLAC program can function.

The first of the three files contains the JFLAC model, called "Model.dat" (see Fig. (B.1)). This file contains the grid node coordinates, as well as the connectivity between the nodes so that JFLAC can internally create the tetrahedron elements. It contains information about the boundary conditions specified for each grid node, and also specifies the initial stress state of each domain element.

The second file, "Material.dat" (see Fig. (B.2)), contains properties of the materials that are assigned to each domain element. These properties include material properties such as Young's modulus, Poisson's ratio, density as well is other inelastic material properties. At this stage it is important to note that each individual tetrahedron of the JFLAC domain can only represent one material, but two neighboring tetrahedrons can have different materials assigned to them. This allows the domain to represent the interaction between different materials.

The last file, "Settings.dat" (see Fig. (B.3)), is a file specifying the settings of the algorithm. For example, it specifies if the algorithm should solve the system by using only elastic material properties, or if failure should be allowed by means of the Mohr-Coulomb condition. It also specifies if MD or if MND should be applied to the system. Other settings that are not relevant to this section are also specified.

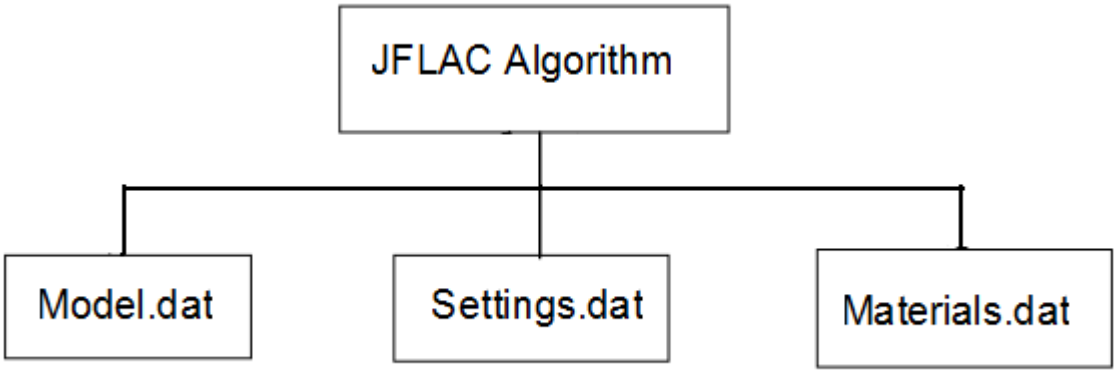


Figure 3.12: Input files of the JFLAC algorithm.

The calculation sequence of the JFLAC algorithm can be divided into several independent components. A flow diagram of the calculation steps involved in the algorithm is illustrated in Fig. (3.13).

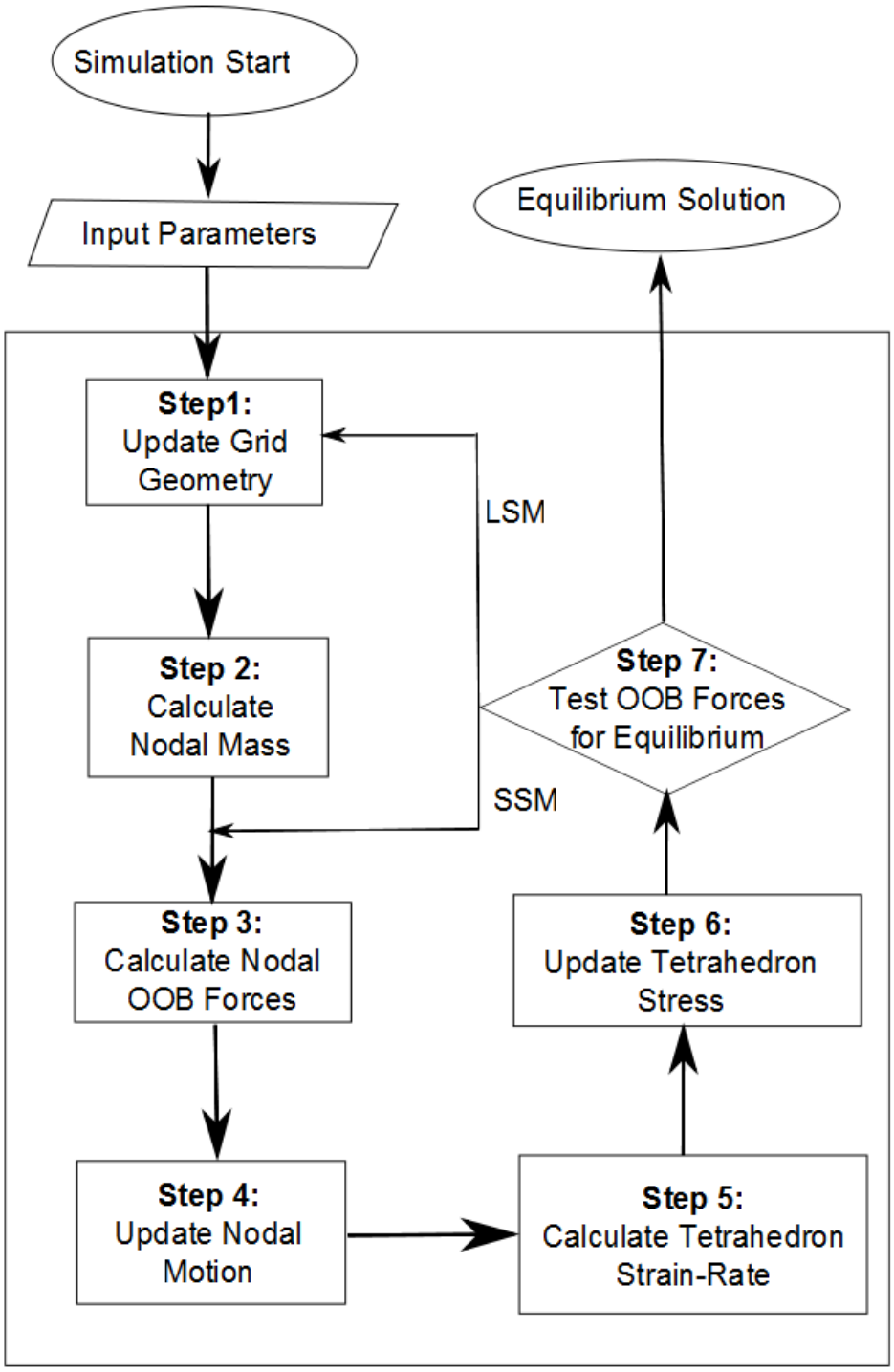


Figure 3.13: Flow diagram of FLAC algorithm.

A basic description of the sequence is as follows:

- **Step 1: Update Grid Geometry** - Node positions are updated by adding the displacement increments calculated in the previous time step. At the initial run time this step can be skipped since displacement increments are assumed to be zero.

- **Step 2: Calculate Nodal Mass** - Tetrahedron masses are calculated and evenly distributed to the four connected grid nodes at their vertices.

- **Step 3: Calculate Nodal Out-of-balance Forces** - Nodal force contributions are calculated from the connected tetrahedron applied loads and body forces.

- **Step 4: Update Nodal Motion** - The equations of motion are invoked to derive new nodal velocities and displacements.

- **Step 5: Calculate Tetrahedron Strain-Rate** - Tetrahedron strain-rates are derived from nodal velocities. If the applied constitutive model allows failure to occur, MD must be applied on the strain-rate if hexahedrons are used in the model or MND must be applied on the strain-rate if only tetrahedrons are used in the model.

- **Step 6: Update Tetrahedron Stress** - New stresses are derived from strain-rates, using a prescribed constitutive law. If the applied constitutive model allows failure to occur, MD must be applied on the stresses if hexahedrons are used in the model or MND must be applied on the stresses if only tetrahedrons are used in the model.

- **Step 7: Test Out-of-balance Forces for Equilibrium** - The maximum out-of-balance force for all the grid nodes in the system is tested against a prescribed threshold. If it is lower than this threshold, the system is in an equilibrium state, otherwise the sequence is restarted.

Once the three input files have been created, then JFLAC internally discritises hexahedron zones into tetrahedrons. A material and initial stress state are prescribed to each tetrahedron. A boundary condition is assigned to all the grid nodes, as specified by the "Model.dat" file, and internal variables are initialized. The calculation sequence is started and the system is solved for its equilibrium solution. The nodal positions are stored throughout the simulation. After each time step, the position for each node is updated by adding the displacement obtained from the calculated strain-increment in the previous time step.

The modal masses in Step 2 are then calculated by taking the average mass of the tetrahedrons connected to it, using Eq. (3.109). Note, that if no tetrahedrons are removed as the simulation progresses, this step does not have to be repeated. The nodal mass values only have to be calculated once and can be stored in memory. However, if tetrahedrons are removed from the system as the simulation progresses, then the masses need to be updated. Once the nodal mass of each grid node is known, Eqs. (3.110), (3.115) and (3.116) are used to calculate the nodal forces, velocities and displacements respectively. At this stage, nodal force damping is applied by means of Eq. (3.122). New tetrahedron strain-rates can then be derived from Eq. (3.9). MD or MND can be applied to the strain-rates if this was specified by the user in the settings file. The tetrahedron strain-rates are used to calculate the elastic stress for all the tetrahedrons by applying Hooke's law, shown in Eq. (3.13). If the user specified that failure is allowed in the domain, the Mohr-Coulomb condition is applied to find the new stress value for each tetrahedron by means of Eq. (3.19), as explained in Section (3.4). MD or MND can also be applied to the tetrahedron stress if it is specified by the user. Finally, if the maximum out-of-balance force of all the nodes in the system is smaller than a prescribed threshold in the "Settings.dat" file, then the system is in equilibrium. If this is not the case, and the strain-rates are small, then the system is assumed to be in SSM and the sequence can be restarted from the nodal mass calculation in Step 2. Otherwise, it is restarted from Step 1.

An important point to remember is that each step in the calculation sequence must finish before the next step can commence. For example, tetrahedron stresses cannot be calcula-

ted if the strain-increments are unknown. This is significant due to a further contribution made to the algorithm by the author, described in Section (4.1).

3.12 Summary

The JFLAC grid and the discritisation thereof into a Lagrangian grid were discussed in detail. The nodal formulations of the governing equations were derived and a detailed description of the algorithm was given.

Chapter 4

Contributions to JFLAC

During the implementation of JFLAC, the author noticed that the algorithm is highly computationally parallelizable. Therefore a second version of JFLAC was created that allows for the use of multiple CPU processors. Later on, these two versions were merged such that if the user specified in the "Settings.dat" file that only a single CPU thread was allowed to perform the simulation, JFLAC would recognize this and perform a part of the code using only a single thread. If however, the user specified that more than one CPU core could be used to perform the simulation, then JFLAC would use a totally different section of code to perform the simulation.

The boundary conditions, as described in Section (3.2), did not always supply the best result accuracy for the simulations when JFLAC was in its testing phase. It was discovered heuristically that the boundaries had to be placed far from areas where high result accuracy was needed. This caused the model to become large and consequently it took large amounts of time to execute. To overcome this, a new boundary condition type, the BNS, was developed to allow for the boundaries to be placed closer to the result areas and reduce the number of elements in the model.

The two contributions introduced above are described in this chapter.

4.1 Multithreading of JFLAC

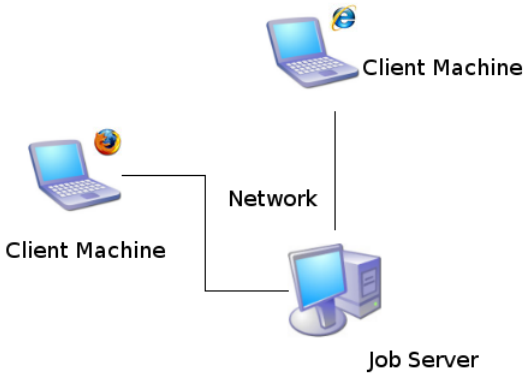
Applications created in the late 20th century were designed to operate on traditional single CPU core machines. However, with the rapid rate of technology improvement, multi core machines became available and software languages improved to meet these changes. Applications therefore have to be structured and designed to utilize the multiple CPU cores. Multithreading is a powerful tool for enhancing the performance of applications, but the design and code structure of multithreaded applications may be more complicated.

The original implementations of FLAC simulation models were developed for single core machines but recently multithreaded versions became available. Theoretical documentation provided with a legal copy of FLAC gave a good understanding of the mathematics and physics involved in the algorithm, but does not mention how to multithread the algorithm. The author coded and implemented JFLAC from this theoretical documentation and created a multithreaded version as an improvement.

4.1.1 Types of multithreading

Currently two types of multithreading exist. The first type is called *Shared Memory* multithreading as illustrated in Fig. (4.1a). With this type of multithreading, an algorithm runs on a single multicore machine and utilizes the resources of only that machine. An example of such a machine is a normal Dual Core laptop with more than one CPU core. The second type of multithreading is called *Server Farm* multithreading as illustrated in Fig. (4.1b). With this type of multithreading, an algorithm is executed on a specific machine known as the *job server machine*. The job server scans a network for other client machines in the same network that has available resources. It then assigns a specific job to a particular machine via the network, and thereby utilizes the client's resources. After its task is completed, the client sends the data back to the job server and combines information from all the connected clients. The server then redistributes new tasks to clients in the network. An example of a server farm is the Google Cluster.

(a) Shared memory multithreading.



(b) Server farm multithreading.

Figure 4.1: Shared memory multithreading.

For the purpose of this thesis JFLAC was implemented using shared memory multithreading on a machine with 8 CPU cores. Each core has a clock speed of 2.3 Giga Hertz and the total memory of the machine is 32 Gigabytes.

4.1.2 Implementing multithreading of the JFLAC algorithm

As mentioned in Section (3.11), it is important to note that a specific step in the calculation sequence cannot be executed unless the previous step in the sequence has finished. For this reason, each individual step must be multithreaded on its own. This implies that if a number of CPU threads is assigned to execute a specific step in the sequence, all the threads have to finish completing their assigned tasks before the next step can commence.

This can cause a thread that performs its task quicker than others to wait until other threads have finished their tasks before it can continue with the next sequence. Although this is not ideal, it was found from experimentation that a 70% speed-up in simulation time was gained for each additional thread used.

Before the steps involved in the multithreading of each calculation step can be explained, a few fundamentals for multithreading the algorithm are discussed.

It is important to keep track of all the grid nodes in the simulated domain as well as all the tetrahedron or hexahedron elements that are connected to the particular grid nodes. At execution of the simulation, a list of all the grid nodes $< N^a >$ is stored in memory, where $<>$ represents a list, a ranges from $1, ..., n$ and n is the total number of grid nodes. A second list that contains all the tetrahedron elements $< T^k >$ is also stored in memory, where k ranges from $1, ..., z$ and z is the total number of tetrahedrons. A connectivity map, *map*, that contains information about the connection between grid nodes and tetrahedron elements is also stored. This is illustrated in Fig. (4.2).

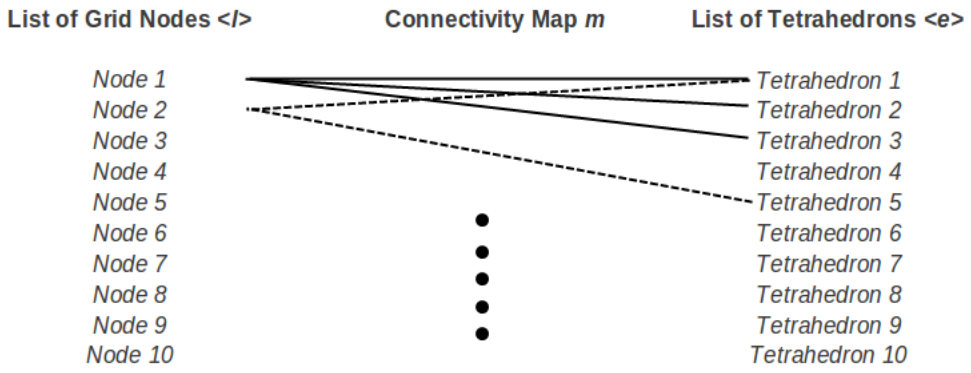


Figure 4.2: Graphical representation of the connectivity between grid nodes and tetrahedrons.

If multiple threads are assigned to perform calculations on the list of grid nodes, the

list is divided into parts equal to the number of threads. For example, consider the model having 10 grid nodes as shown in Fig. (4.2) and two threads are assigned to perform certain calculations on the list. The list $< N^a >$ is then divided into two segments and calculations on this list are assigned independently to the two individual threads, as illustrated in Fig. (4.3). Each thread performs calculations on its 5 assigned nodes.

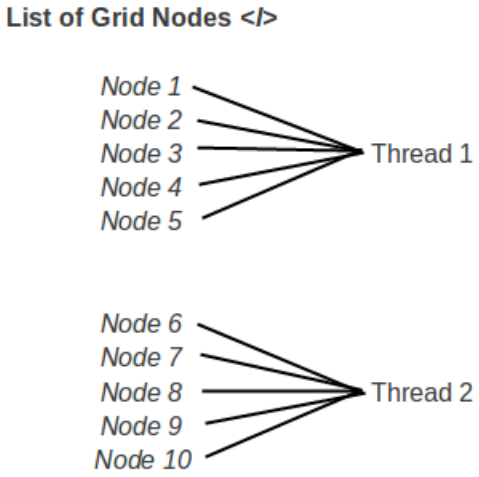


Figure 4.3: List of grid nodes divided for multithreading.

If 8 threads are used to perform calculations on this list of 10 nodes, then the list will be divided into 8 segments and two of those segments will contain an extra entry. In this case, it becomes more evident that the threads containing lists with the least entries will finish their calculations faster than the threads delayed by larger lists. With the understanding of the multithreading scheme explained above, more detail is given on how each part of the algorithm calculation sequence is performed.

It is assumed that g threads are assigned to the multithreading task. When a list is divided into g segments, it implies that the list is divided into g parts and assigned to an appropriate thread.

4.1.2.1 Multithreading the nodal geometry update calculation

For this step in the sequence only the list $< N^a >$ is needed. The model geometry is updated by moving the grid nodes to their new positions by adding the displacement increments calculated in the previous time step and obtaining new positions. The list $< N^a >$ is divided into k segments to perform the addition operation.

Note that if the algorithm is executed in SSM, the nodal geometry update task may be skipped.

4.1.2.2 Multithreading and implementation of nodal mass calculation

Lists $< N^a >$ and $< T^k >$ are needed for this calculation process and the connectivity map m is used to find node-element connectivity. The list $< N^a >$ is divided into g segments. For a specific node in $< N^a >$ assigned to a thread, *map* is used to find the connected elements in $< T^k >$. For each connected element the internal mass is calculated and stored.

This step in the sequence can be skipped if the simulation is in SSM.

4.1.2.3 Multithreading and implementation of nodal force calculation

Lists $< N^a >$, $< T^k >$ and *map* are needed to perform this task. The list $< N^a >$ is divided into g segments and for each node in $< N^a >$, *map* is used to find the connected elements in $< T^k >$. For each node the out-of-balance force is calculated from Eq. (3.110). Eqs. (3.115) and (3.116) are used to calculate the nodal velocities and displacements. This task is the first computational expensive task in the algorithm and large performance increases can be gained from multithreading.

4.1.2.4 Multithreading and implementation of the strain-rate calculation

Once the nodal velocities for all the nodes in $< N^a >$ are calculated, then it can be used to calculate the strain-increments for each tetrahedron element using the difference formulation in Eq. (3.9). List $< T^k >$ is divided into g segments and *map* is used to

find the node-tetrahedron connectivity in $< N^a >$. Once all the threads are finished calculating the tetrahedron strain-increments, then they are re-assigned to perform MD or MND on $< T^k >$, if the user specified such in the "Settings.dat" file.

4.1.2.5 Multithreading and implementation of the stress-rate calculation

Implementation of multithreading the stress-rate calculation is similar to the process described in Section (4.1.2.4).

4.2 Implementation of the Boundary Node Shell boundary condition

As mentioned earlier, FLAC is an example of a domain method for modelling the state of deformation and stress in a solid. As a consequence, all domain methods model some finite body or a finite part of some larger, perhaps infinite, body. Whenever this is the case, one needs to take into account the fact that at least a part of the model boundary is not a real free surface, but a virtual separator between the modelled material and its surroundings. The question arises whether one needs to take into account the interaction between the material in the model domain and the surrounding material. If one chooses to ignore these interactions, then the results for the stress and the deformation within the modelled domain will be inaccurate. Yet, one may choose to ignore this problem if the points where high result accuracy is needed, are sufficiently far from the virtual boundary.

Since the interaction between a body and its surroundings is conducted through the common boundary, the distortion in the elastic state owing to this interaction will decrease sufficiently before reaching the points of interest. Unfortunately, making sure that the virtual boundary is sufficiently far from the points of interest, is not always practical since it leads to a significant increase in the model size and can be very demanding on computer resources. The alternative is to keep the model as small as possible and to account for the fact that at least a part of the boundary is not real but virtual. This can be done by computing additional forces which originate from the surrounding material and are the reaction to the processes taking place inside the modelled domain. One possible way of

doing this is by adding a Boundary Nodes Shell (BNS) to the model.

The BNS is a tool for computing the forces due to the response of the surrounding material to any changes in the common boundary. These forces will be computed at a set of points called sources. The sources are placed outside the modelled domain in an assumed infinite linear elastic material. At the same time, the virtual movements of a set of reference points on the boundary are monitored. Fig. (4.4) illustrates the placement of these points.

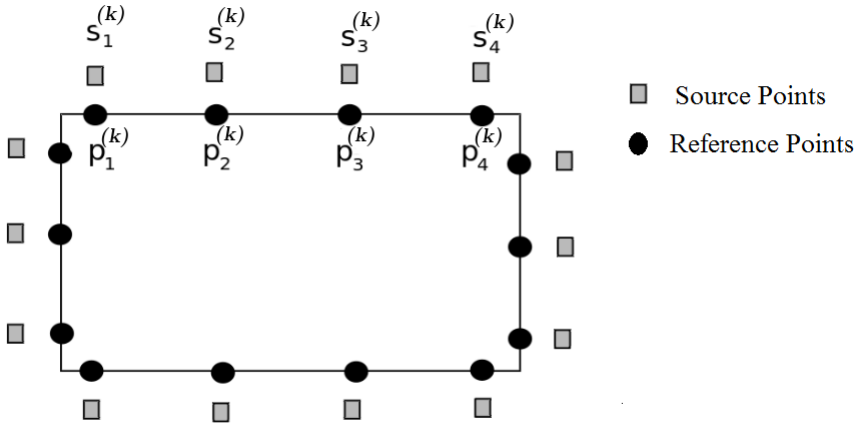


Figure 4.4: The placement of source points and target points to perform the BNS.

The idea of the BNS is to find a set of *fictitious forces* acting at the sources that would compensate the virtual movements of the reference points on the boundary. As a result the boundary of the modelled domain will remain unchanged, but a set of new forces will act on it simulating the reaction of the surrounding to any change of state inside the domain.

To describe how the BNS works, assume that for any time step in the system which is not in its equilibrium state, the stress is calculated. A set of reference points are placed between the modelled domain and the surrounding material area identified as the boundary nodes of the grid and labeled:

$$p_i^{(k)} = (x_1^{(k)}, x_2^{(k)}, x_3^{(k)}) \quad , \quad k = 1, 2, ...n_b$$

where n_b is the total number of boundary nodes in the system. Source points, labeled

$$s_i^{(k)} = (y_1^{(k)}, y_2^{(k)}, y_3^{(k)}) \quad , \quad k = 1, 2, ...n_b,$$

are placed in the outer material. In the simplest implementation of the BNS, the source points are as many as the reference points, and, as a simple rule of thumb, are placed twice as far away from the boundary as the size of the largest tetrahedron in the domain. One might say that the displacement of the boundary nodes causes the outside elastic material points (source points) to shift. Instead of shifting these points, one solves a system of equations for the components of fictitious forces acting at the sources and being capable of causing exactly the opposite displacements as those proposed by the target points. If a concentrated body force

$$B_i^{(k)} = (B_1^{(k)}, B_2^{(k)}, B_3^{(k)}) \tag{4.1}$$

is acting at source $s_i^{(k)}$, the displacement, $u_i^{(k,m)}$, that will induce at reference point, $p^{(m)}$, $with\ m = 1, 2, ..., n_b$, will be

$$u_i^{(k,m)} = \sum_{j=1}^{3} U_{ij}(s^{(k)}, p^{(m)}) B_j^{(k)} \tag{4.2}$$

where the kernel $U_{ij}(s^{(k)}, p^{(m)})$ is the Kelvin solution for the displacement due to a unit concentrated force [17]. The total displacement induced at a given reference point due to the unknown forces, $F_j^{B^{(k)}}$, at all the source points will be the sum of expressions such as the one in Eq. (4.2), in accordance with a superposition principle. Inserting the virtual displacements at the reference points in the left-hand side of Eq. (4.2), one arrives at a system of linear equations for the components of the fictitious forces at the sources. Once the system in Eq. (4.2) is solved and the fictitious forces have been found, one can compute the stress these fictitious forces would induce at the reference points, $p^{(k)}$. The next step is to use the found fictitious forces to compute the corresponding tractions $t_j^{(k,m)}$ at the reference points. The traction on reference point $p^{(m)}$ induced by the fictitious force acting at source $s^{(k)}$ is given by:

$$t_i^{(k,m)} = \sum_{j=1}^{3} T_{ij}(s^{(k)}, p^{(m)}, \mathbf{n}^{(m)}) F_j^{B^{(k)}} \tag{4.3}$$

[17] where $T_{ij}(s^{(k)}, p^{(m)}; \mathbf{n}^{(m)})$ is the traction kernel which is obtained by differentiating the Kelvin solution and $\mathbf{n}^{(m)}$ is the outward force normal to the boundary at the reference point. These tractions are used to update the boundary conditions of the grid nodes on the virtual boundary. As a result of this procedure the modelled domain retains its boundary unchanged but updates the boundary conditions after each time step, thus taking into account the reaction of the surrounding material.

The BNS boundary condition gives a more physically correct response than both the displacement and stress boundaries described in Section (3.2). The major drawback is it increases the simulation time. The system in Eq. (4.2) and Eq. (4.3) must be solved for every time step which in itself is a time consuming process. Also, larger domains and finer discritisation of these domains, result in a significant increase in the number of grid nodes on the boundaries. This significantly increases the number of reference points in Eq. (4.2) and makes it considerably more difficult to find a solution for the fictitious forces.

As mentioned in Section (3.2), a combination of boundary conditions can be used in the simulation. By choosing the BNS condition for boundaries that are close to an area where high result accuracy is needed, in conjunction with displacement or stress boundary conditions for the boundaries that are placed further away, can speed up the system significantly.

4.3 Summary

The author described the implementation a multithreaded version of JFLAC. This implementation showed a significant improvement (approximately 70%) in the simulation time for each additional CPU core used to perform the simulation. The BNS boundary condition and the implementation thereof were also given.

Chapter 5

Results and discussion

This chapter describes the three case studies that were performed with JFLAC during its testing phase. First of all, as with most modelling software packages in development, the reliability of JFLAC was tested against a well-known analytical solution. During the analytic test, the results obtained by JFLAC were also compared against those obtained by FLAC. Once a good comparison between JFLAC, FLAC and analytical solutions were found, then JFLAC was used to conduct a case study for a platinum mine, located in the Bushveld complex in South Africa. A final case study was investigated where the application of the BNS boundary condition was compared to Salamon's analytic solution [17], as well as the results obtained by using fixed displacement- or stress boundary conditions.

5.1 Analytical solution - cylindrical hole in an infinite Mohr-Coulomb material

The analytic solution for a cylindrical opening in an infinite Mohr-Coulomb material is known for a body under a hydrostatic stress field. It provides the exact solutions for the stresses around the cylindrical opening inside a solid. This verification example is also contained in [10], and provides a good basis for comparing the results obtained by JFLAC against FLAC. Some images of stress and displacement results obtained with FLAC, were taken from the FLAC manual and used in this section.

Imagine that a cylindrical opening is made in a Mohr-Coulomb material. For now, the material properties are unimportant. Let the radius of the opening be denoted by R and the radius of the fracture zone (yield zone radius) be denoted by R_f. Let the hydrostatic stress at an infinitely far distance from the opening be denoted by q. The analytic solutions for the stress in polar coordinates, as given by Ryder and Jager [17] are

$$\sigma_r = s_c^{'} \left(\frac{r}{R}\right)^{N_\phi - 1} \tag{5.1}$$

and

$$\sigma_\theta = N_\phi s_c^{'} \left(\frac{r}{R}\right)^{N_\phi - 1} \tag{5.2}$$

where $[R < r < R_f]$ inside the fracture zone and $s_c^{'}$ is known as the effective support resistance. This comprises of the actual support resistance s_c and the UCS, σ_c, of the rock and is given by

$$s_c^{'} = s_c + \frac{\sigma_c}{N_\phi - 1}. \tag{5.3}$$

Parameters for N_ϕ and σ_c can be found from Eqs. (2.32) and (2.33) in Section (2.2) and are again given:

$$N_\phi = \frac{1 + \sin\phi}{1 - \sin\phi} \tag{5.4}$$

and

$$\sigma_c = 2C_0\sqrt{N_\phi} \tag{5.5}$$

where C_0 is the material cohesion and ϕ is the material friction angle. Outside the fracture zone $[r > R_f]$ the solution for the stresses are expressed as

$$\sigma_{r,\theta} = q \pm \frac{(N_\phi - 1)q + \sigma_c}{N_\phi + 1}\left(\frac{R_f}{r}\right). \tag{5.6}$$

An exact value for the yield zone radius is also given as

$$R_f = R\left[\frac{2q - \sigma_c}{(N_\phi + 1)s_c'}\right]^{\frac{1}{N_\phi - 1}}. \tag{5.7}$$

5.1.1 The JFLAC model

The material properties and stress values as used in the FLAC verification manual were used in the JFLAC model. A constant hydrostatic stress of 30 MPa was assigned to a solid model. A cylindrical opening, with a radius of 1 m, is made in the solid. The far x, y and z boundaries were placed at a distance of five hole-diameters from the axis of the hole and a constant force, **F**, calculated from the hydrostatic stress, was placed on the boundary nodes of the grid. Fig. (5.1) shows a 2D picture of the JFLAC model that was generated.

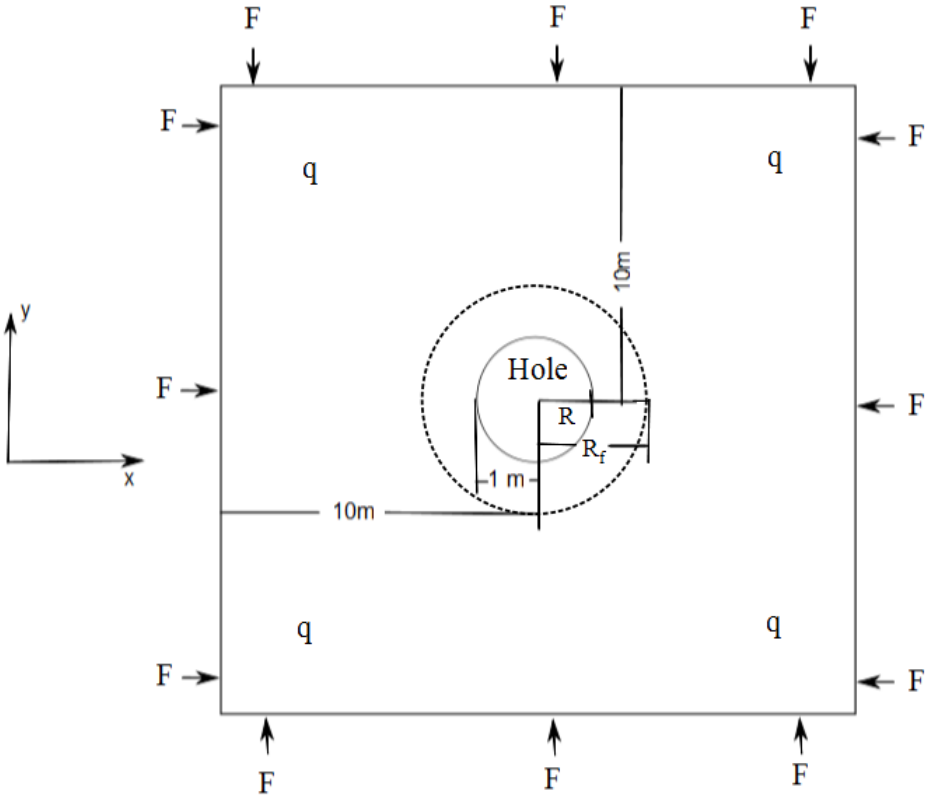


Figure 5.1: The 2D representation of the model used in this case study.

A material, with properties as listed in Table (5.1), is assigned to each tetrahedron in the JFLAC domain.

Table 5.1: Mohr-Coulomb material properties used to compare with the analytical solution.

Shear modulus (G)	2.8 GPa
Bulk modulus (K)	3.9 GPa
Cohesion (c)	3.45 MPa
Friction Angle (ϕ)	30^o
Dilation Angle (ψ)	30^o

5.1.2 Results found by FLAC

The results in Fig. (5.2) are taken directly from the FLAC Verification manual. This shows results obtained by FLAC for σ_r and σ_θ on any line of points going outward from the excavation. The solid and dashed lines indicate the analytic σ_r and σ_θ respectively. The $\bullet$ and $+$ shows the results obtained by FLAC for σ_r and σ_θ respectively. Fig. (5.3) shows contours of displacement near the hole.

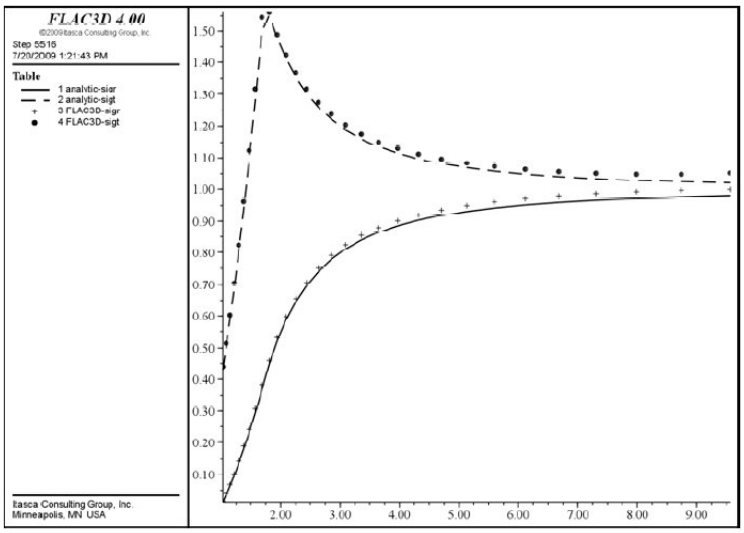


Figure 5.2: Radial and tangential stress results obtained by FLAC, compared with analytical values.

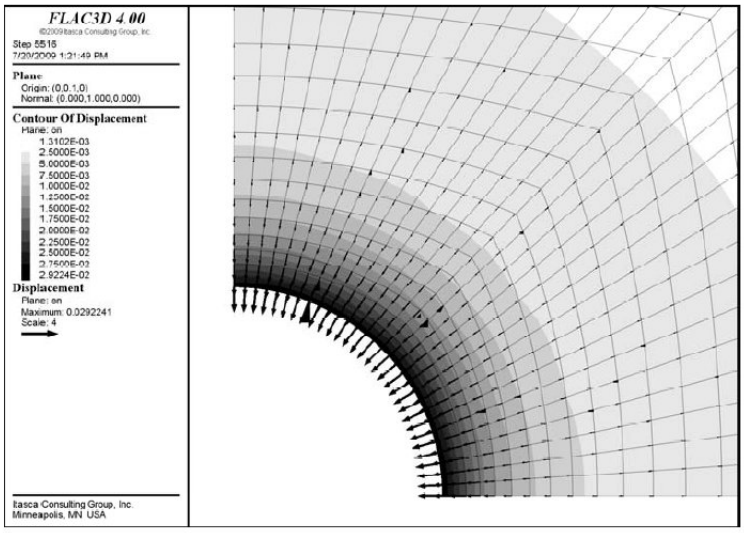


Figure 5.3: Displacement contours obtained by FLAC near the hole.

5.1.3 Results obtained by JFLAC

The JFLAC model in Fig. (5.1) consisted of 3 600 hexahedron zones and since DGD was applied to the model, such that it contained 36 000 tetrahedrons. MD was applied to each hexahedron element.

Fig. (5.4) shows the displacement values for the grid nodes obtained at the equilibrium solution (warmer colors indicate a larger displacement). Note the fine discritisation of nodes around the opening and how discritising the nodes get more sparse as the nodes progress further away from the boundary. The maximum displacement is 0.028 m at the edge of the opening. This result is similar to the maximum displacement in the FLAC results of Fig. (5.3). Fig. (5.5) shows the direction of nodal displacement with vector lengths the magnitude of the displacement.

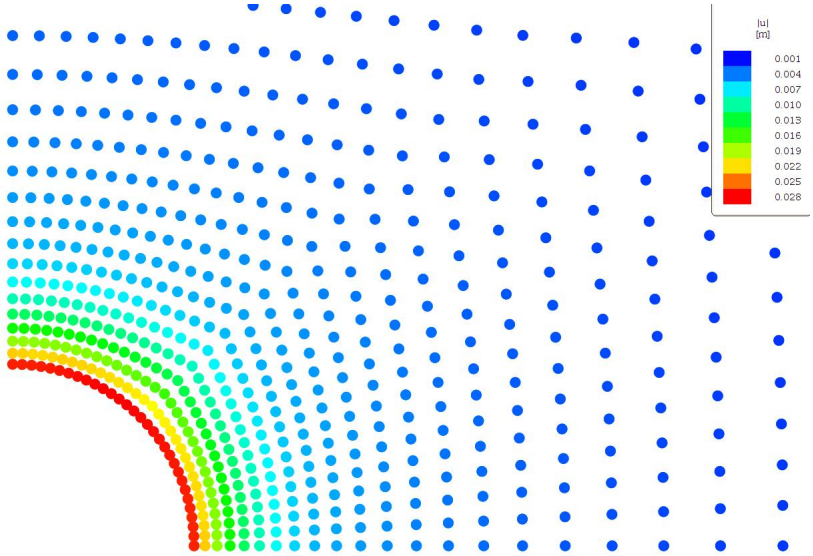


Figure 5.4: Displacement contours obtained by JFLAC.

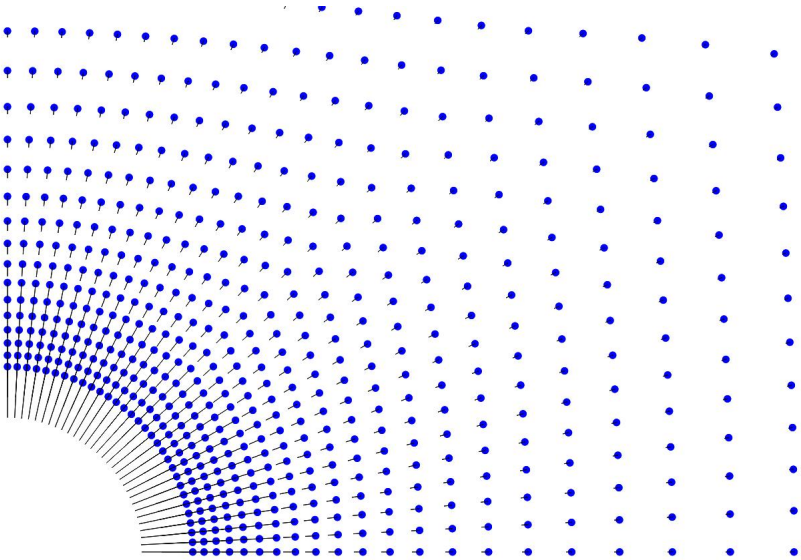


Figure 5.5: Displacement vectors obtained from JFLAC on the grid nodes.

Fig. (5.6) shows the areas around the opening where elements have failed (indicated by smaller dots) according to the Mohr-Coulomb condition. This also gives an idea of the length of the yield zone. The yield zone radius in this example is found to be 1.73 m, which corresponds well to the theoretical value of 1.74 m (From Eq. (5.7)).

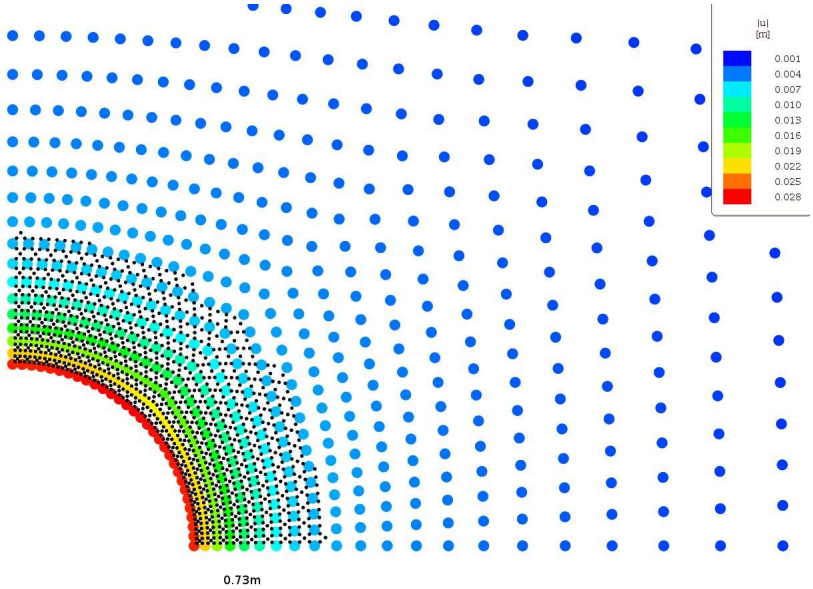


Figure 5.6: Failed elements indicating the yield zone radius.

Fig. (5.7) shows the comparison between JFLAC and the analytic solutions for σ_r and σ_θ. The solid red line shows the analytic solution for σ_θ while the solid green line shows the analytic solution for σ_r. A scatter plot of the σ_r and σ_θ stresses of all the tetrahedrons is also shown. Blue + indicates σ_θ values and magenta $\times$ values indicates σ_r obtained by JFLAC. A 2.03% average deviation (error) from the analytical solution was calculated from the stress results obtained by JFLAC. This compares very closely to the 2.0% error stated in the FLAC Verification manual.

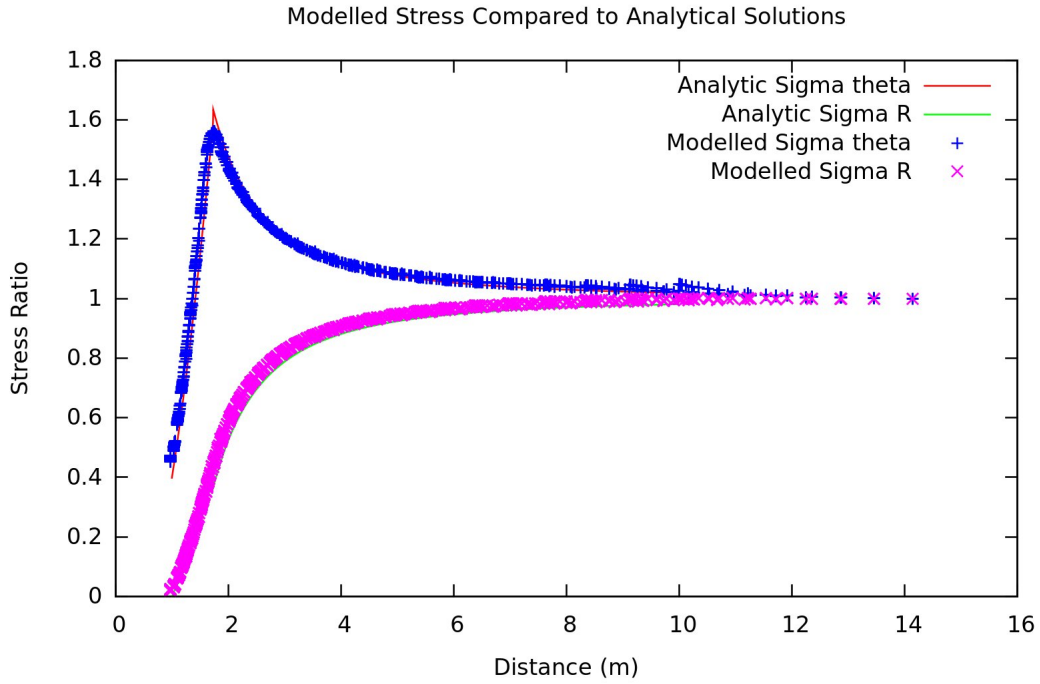


Figure 5.7: JFLAC comparison to analytical solutions.

5.1.4 Summary

JFLAC produced good results and showed a good comparison with FLAC. Both methods showed an approximate average deviation of 2% from the analytical solution. This case study proves that JFLAC is stable and accurate enough to conduct further case studies.

5.2 Case Study 1 - stress influence of a large underground excavation on nearby tunnels

This case study was conducted on a platinum mine in the Bushveld Complex (BC). A basic background on the geology of the BC as described by Ryder and Jager [17] is given, followed by a detailed discussion of the problem.

The main body of the BC is the largest known layered intrusion in on earth and is the result of a large magmatic event that occurred approximately 2 billion years ago. It consists of basic igneous rock types that are intruded by granophyres and granites. The extent and shape of the main platinum reef take the form of an ellipse which has a 350 km major axis oriented East-West, and 150 km minor axis, oriented North-South. The BC consists of four main zones: the Main zone, the Upper Critical zone, the Lower Critical zone and the Lower zone. The upper and lower critical zones include the most important mining horizons of the BC and two seams, or reefs, namely the Merensky reef and the UG2 reef, have been exploited and have been mined almost exclusively in the western lobe of the complex.

This case study was performed on a platinum mine in the BC. Both the Merensky- and UG2 reefs are being mined extensively. Merensky mining was performed in the earlier 20th century while UG2 mining only started in the late 20th century. Both ore bodies have an azimuth of 45 degrees and plunges by 20 degrees. In some cases the middling (distance between the two reefs) is as little as 20 m and the depth at the deepest point in the mine ranges over 2000 m. The small middling could lead to a potential problem as excavations on both reefs could *influence* each other and cause these areas to be seismically hazardous. Virgin stress levels at this depth ranges around 60 MPa. Fig. (5.8) illustrates outlines of mining areas on both reefs - red areas indicate mined out areas on the Merensky reef and green areas indicate mined out areas on the UG2 reef, below the Merensky reef.

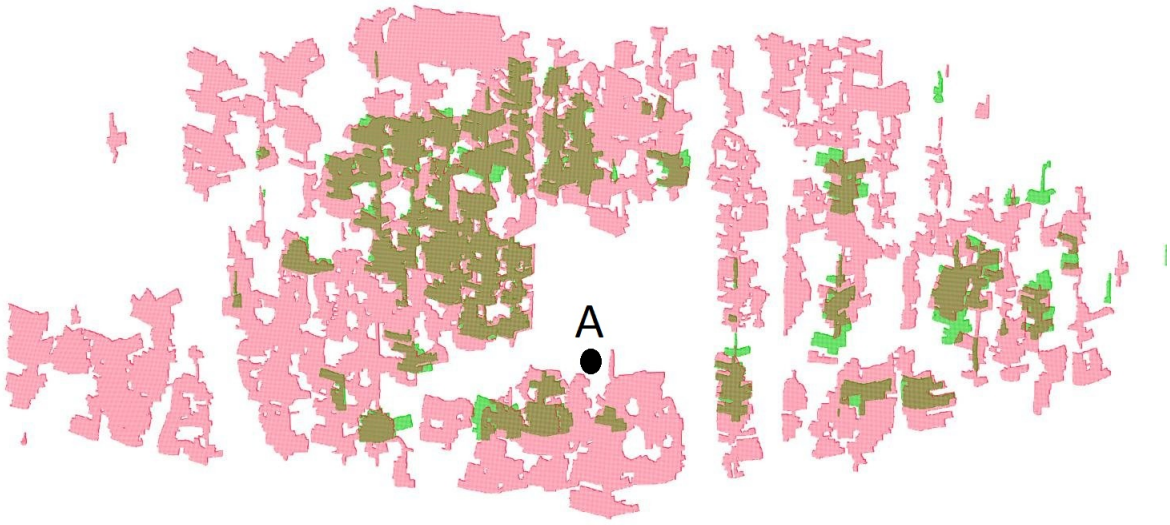


Figure 5.8: Merenski and UG2 mined out areas.

The area of interest in this case study is at point A in Fig. (5.8). A large excavation is to be constructed in this area. Fig. (5.9) is a zoomed in view of Fig. (5.8). Grey lines indicate existing tunnels and blue areas indicate the the areas where the excavation is to be constructed.

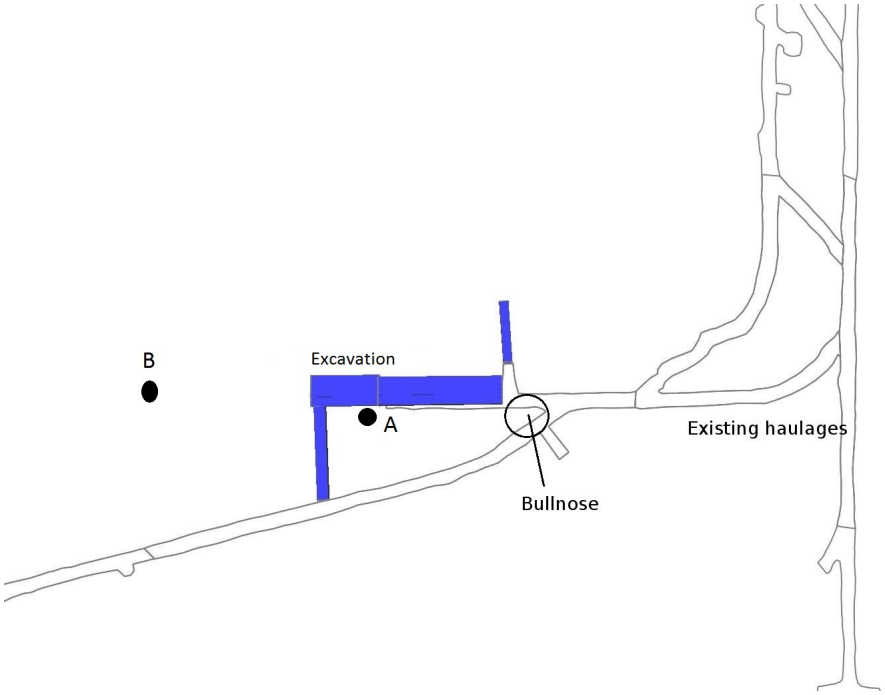


Figure 5.9: Zoomed in view of the refrigeration plant area.

The main concern of excavating the rock mass for this region is the influence it will have on the bullnose (BZ) in Fig. (5.9). A BZ is an area where excavations form a sharp corner in the remaining solid rock mass. If stress levels in this area are high, the BZ could easily get damaged and the surrounding areas can become hazardous. The main areas that were investigated include:

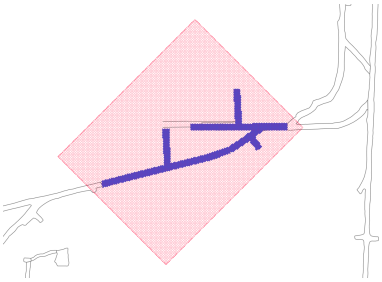
1. Determining the stress level inside the BZ in Fig. (5.9) before excavation for the refrigeration plant.

2. Determining the effect of the excavation at point A on the stress level inside the BZ.

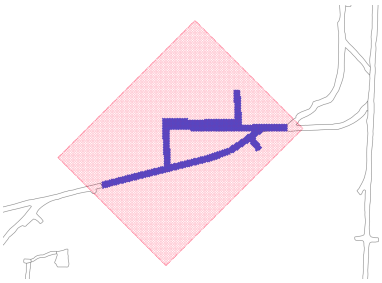
3. Determining if it would be more favorable to the BZ to place the excavation of the plant at point B.

5.2.1 Simulation input parameters

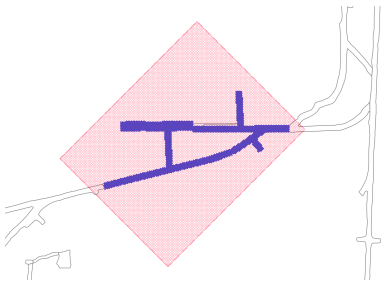
Three model domains were generated, each of which consisted of approximately 1.2 million hexahedrons. Since Mixed Discritisation was applied to all the models, the total number of tetrahedrons in the models was close to 12 million. The first model (Model A in Fig.(5.10a)) describes the basic mined out geometry of the existing tunnels without the refrigeration plant. The second model (Model B in Fig.(5.10b)) describes the current excavations including the planned location of the refrigeration plant as illustrated at point A of Fig. (5.9). The third model (Model C in Fig.(5.10c)) consists of current excavations and includes the refrigeration plant located at point B in Fig. (5.9). Fig. (5.11) is given to provide a better understanding of the model and the location of the excavations with respect to the boundaries of the domain. Blue areas indicate the mined out excavations and red dots indicate the boundaries of the JFLAC domain.



(a) Plan view of Model A



(b) Plan view of Model B



(c) Plan view of Model C

Figure 5.10: JFLAC models for Case study 1.

Displacement boundary conditions were assumed for this model and each hexahedron had a constant prescribed stress (in MPa), calculated from the virgin stress at 2000m below surface, given by

$$\sigma_{virgin} = \begin{bmatrix} 31 & 0 & 0 \\ 0 & 31 & 0 \\ 0 & 0 & 62 \end{bmatrix}. \tag{5.8}$$

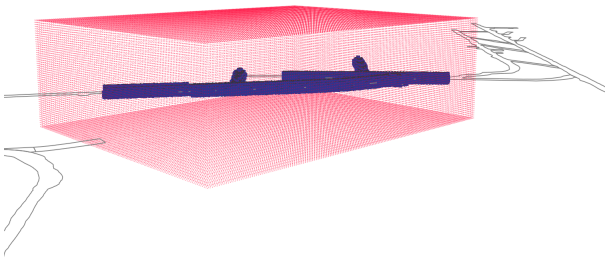


Figure 5.11: 3D view of Model A

Force boundary conditions, calculated from Eq. (5.8) on the surface of the boundary, were placed at the boundary nodes at the start of the simulation. All simulations were performed for a pure elastic material, as well as the Mohr-Coulomb constitutive model. Material properties used for the simulations are listed in Table (5.2).

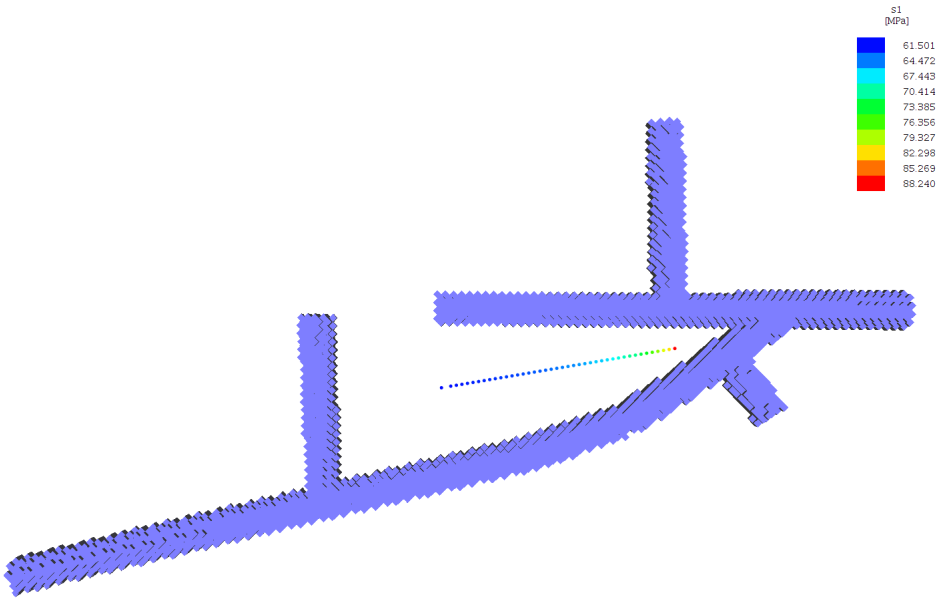
Table 5.2: Mohr-Coulomb material properties for Case study 1.

Shear modulus (G)	28 GPa
Bulk modulus (K)	46 GPa
Cohesion (c)	5 MPa
Material Density (ρ)	2700 kg/m^3
Friction Angle (ϕ)	30^o
Dilation Angle (ψ)	30^o

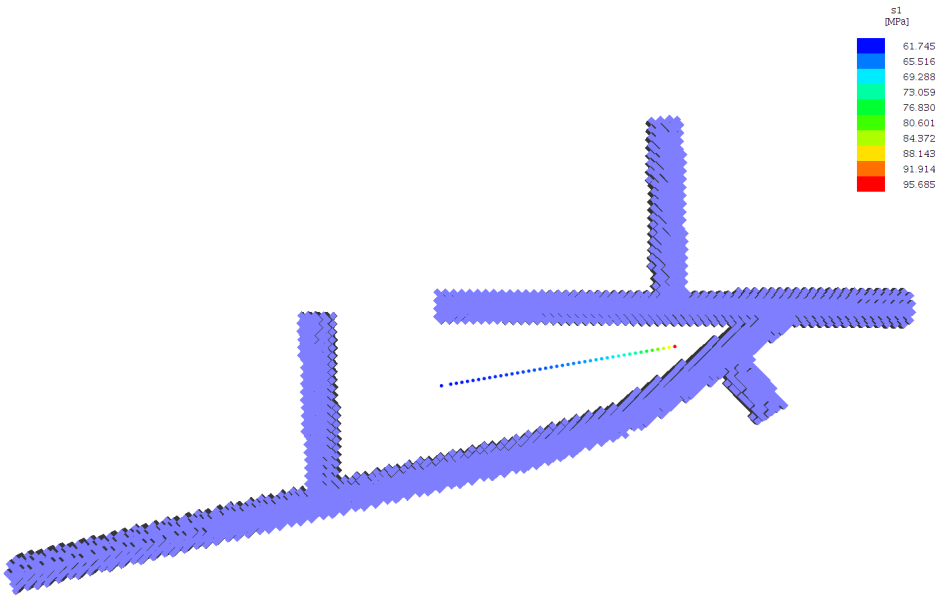
5.2.2 Results

5.2.2.1 The stress level inside the BZ before excavation for the refrigeration plant.

The major principal stress, σ_1, was used as an estimate of the stress level inside the BZ. The results were analyzed on a line that runs through the BZ in Model A as indicated by Fig. (5.12).



(a) Elastic stress results on line.



(b) Mohr-Coulomb stress results on line.

Figure 5.12: Model A stress results.

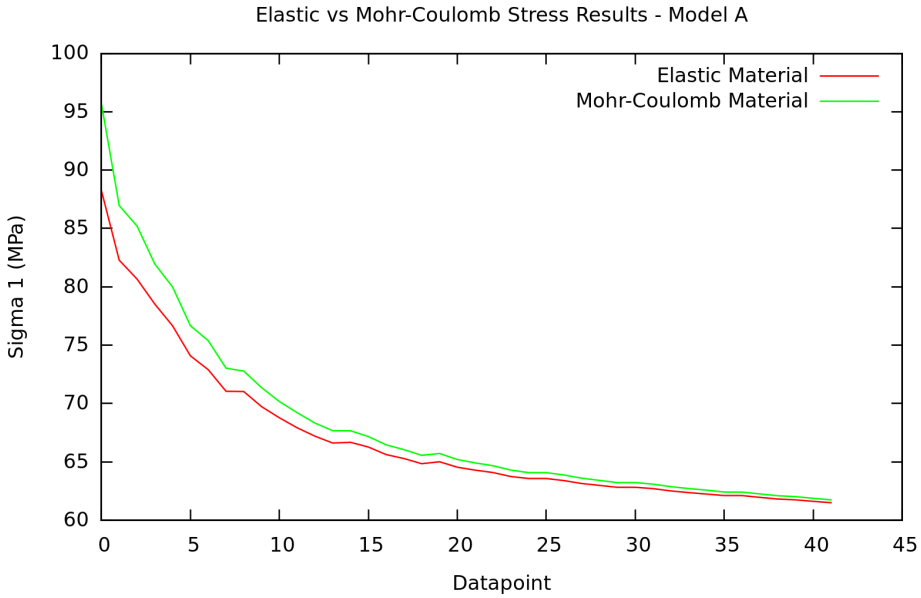


Figure 5.13: Elastic material vs Mohr-Coulomb material results.

Stress values for the elastic material of Model A in Fig. (5.12) indicate values of approximately 88 MPa, while stress results for the Mohr-Coulomb material indicate values of approximately 97 MPa. Fig. (5.13) shows a graphical comparison between the two models. It can be concluded from the above results that a reasonable estimate for the stress level inside the BZ is between 88 MPa and 97 MPa for the material properties in Table (5.2).

Fig. (5.14) shows σ_1 stresses on a cross-section that intersect the BZ. This indicates, as to be expected, that stress levels above and below the mined out excavations are de-stressed while areas on the sides of the excavations, especially the BZ area, are more stressed. Fig. (5.15) shows the extent of the fracture zone around the excavations as a result of the Mohr-Coulomb material. The BZ does not indicate extensive fracturing, except in the sharp corner where the two tunnels meet, as to be expected.

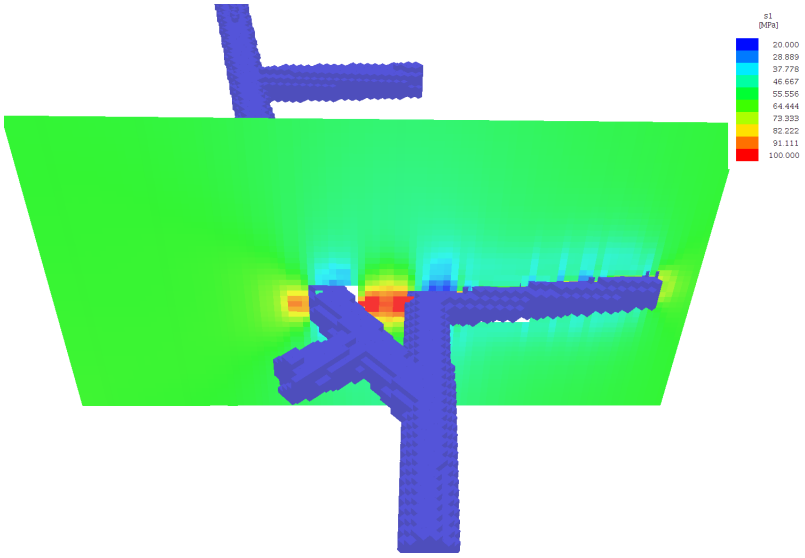


Figure 5.14: Stress on a cross section that intersects the bullnose.

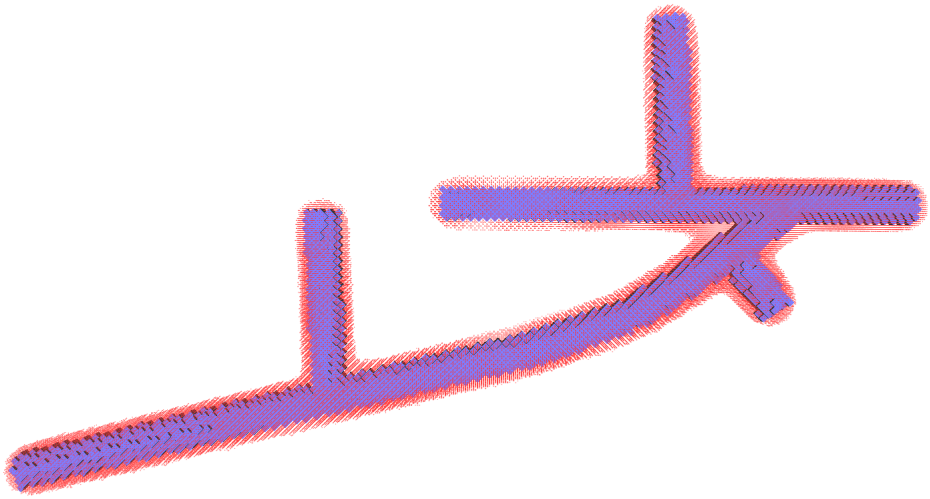


Figure 5.15: Fracture zone around excavations.

5.2.2.2 The effect of the excavation at point A on the stress level inside the BZ

Fig. (5.16) shows a comparison in the stress level on the line inside the BZ between Model A and Model B using a Mohr-Coulomb material.

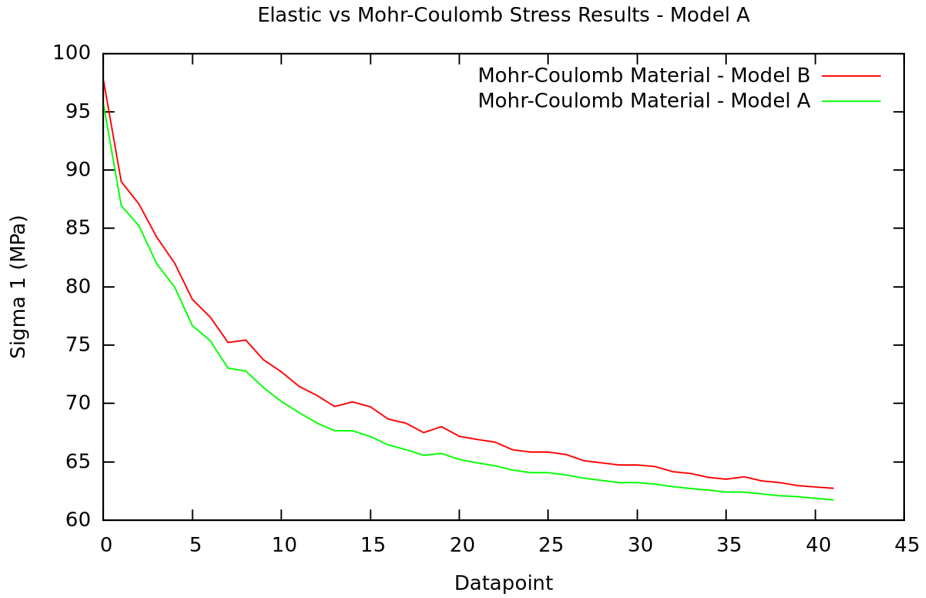


Figure 5.16: Comparison in Stress Level between Model A and Model B.

The trends in Fig. (5.16) show a slight difference in the stress level in the BZ between Model A and Model B. However, this is very small and is debatable whether the difference is large enough to have a significant impact. Fig. (5.17) shows the fracture zone of Model B after constructing excavation. It shows that the BZ is more fractured than the fracture zone of Model A in Fig. (5.15). The larger fracture zone around the excavation extends well into the BZ and could indicate that Model B could be more hazardous than Model A. In this case using the fracture zone instead of the stress level inside the BZ could be a better estimate of the potential hazard of the BZ.

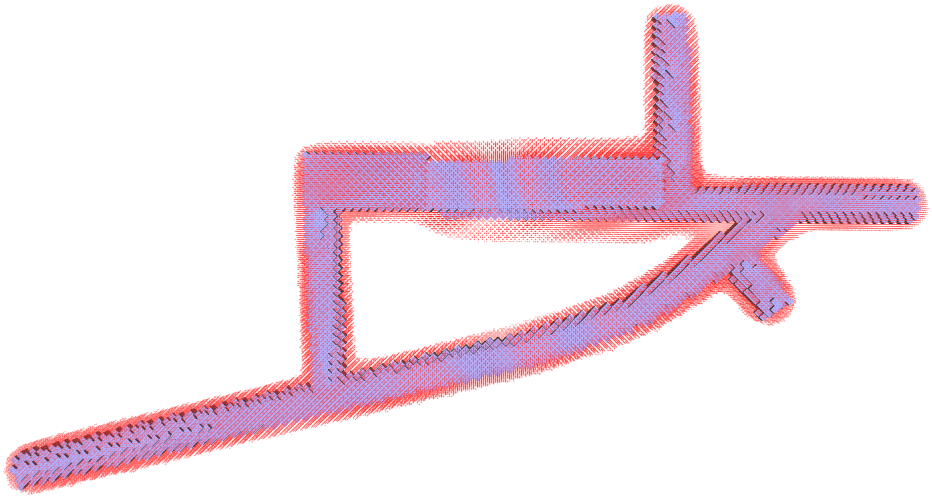


Figure 5.17: Fracture zone of Model B.

5.2.2.3 Appraisal of point B for the refrigeration plant

Fig. (5.18) shows a comparison in the stress level on the line inside the BZ between Model A, Model B and Model C. This shows that Model B causes the stress inside the BZ to be slightly higher that that of Model A and Model C. The stress results of Model A and Model C are practically similar, which means that the excavation has no impact on the BZ. Fig. (5.19) shows the fracture zone of Model C. Although there is a larger fracture zone around the excavation, it does not extend into the BZ area.

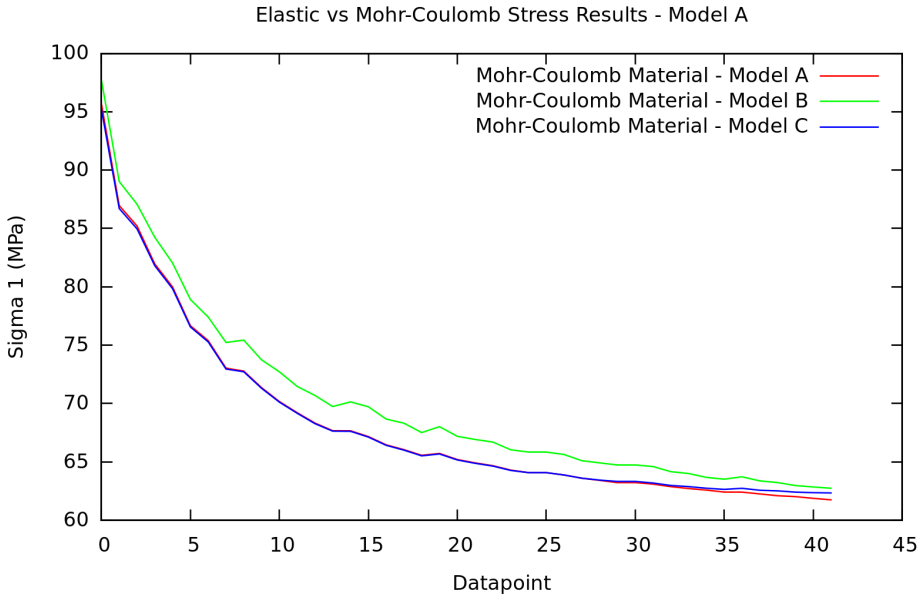


Figure 5.18: Stress comparison between Model A, Model B and Model C.

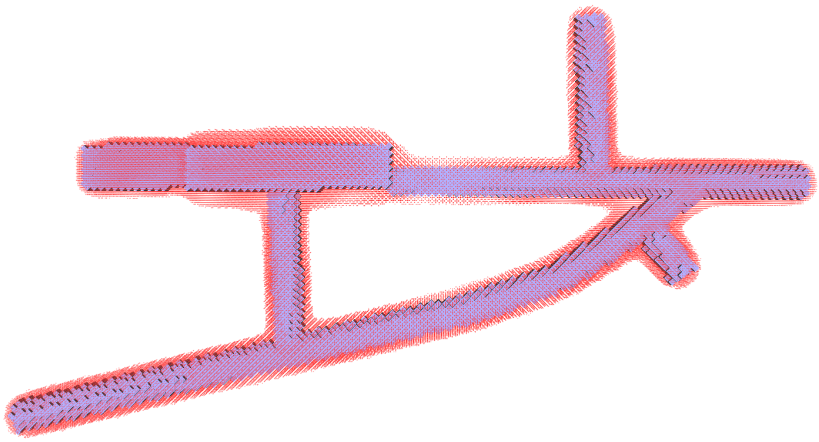


Figure 5.19: Fracture zone of Model C.

5.2.3 Summary

By using JFLAC it is possible to obtain a reasonable estimate of the stress level inside the BZ. Model B shows a slightly higher stress level inside the BZ than those of Models A and C. However, this difference is small and the impact of this difference might be negligible. The fracture zone in Model B extends into the BZ and can indicate that it could be a more hazardous configuration than those of Models A and C. The results of Model C show that it has no impact on the BZ. Stress levels are very similar to those of the current state of Model A and the fracture zone of the refrigeration plant does not extend into the BZ. In this case, it can be concluded that placing the refrigeration plant at position B in Fig. (5.9) will be more favorable than the original position at point A.

5.3 Case Study 2 - an investigation into the Boundary Node Shell

A tabular opening is formed when a thin layer of material is removed from a solid. In the South African gold mining industry, mining takes place on a single planar ore body and has a narrow opening, typically in the order of 1 m. These excavations can span over large distances. Salamon [18] was able to derive the analytic solution for the vertical stress in the plane of excavation as a result of the tabular opening. This solution applies for a homogeneous isotropic elastic medium as well as transversely isotropic and frictionless-laminated elastic strata. The solution for a homogeneous isotropic medium is discussed here and the vertical stress, σ_{zz}, in the plane of the excavation will be analyzed.

Assume that a uniform virgin stress, σ_v, acts on a horizontal tabular excavation. The excavation is assumed to be infinitely thin and has a short side length of 2L, while the long side can be infinitely long. The vertical stress σ_{zz} at any point in the plane of the excavation is expressed as [17]

$$\sigma_{zz} = \sigma_v \sqrt{\frac{x^2}{x^2 - L^2}}, \tag{5.9}$$

where x is the distance from the point to the excavation. This model is illustrated in Fig. (5.20).

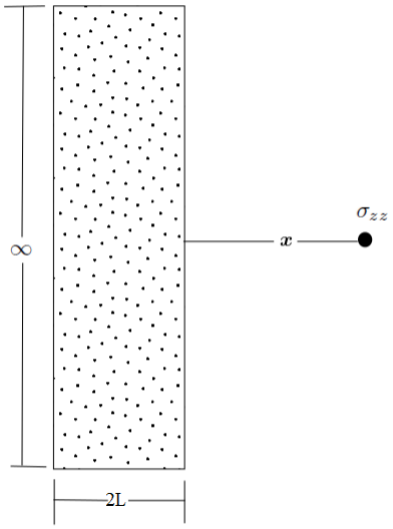


Figure 5.20: Model for Salamon's solution.

A simple JFLAC model, shown in Fig. (5.20), was constructed to compare the performance of the BNS method against the analytical solution for this problem, and also to compare it against the fixed displacement- and stress boundary conditions. The model has a dimension of 100 m x 100 m x 100 m in a Cartesian coordinate system. An excavation of dimensions 10 m x 90 m x 10 m was made in the center of the model. Three simulations were performed for each of the boundary condition in question, and vertical stress results were calculated for a line of points along x in Fig. (5.20).

Fig. (5.21) shows the results obtained by this numerical experiment for each of the three simulations as well as the analytic results obtained from Eq. (5.9). The stress obtained by using BNS boundary conditions lies between the results obtained using fixed displacement- and stress boundary conditions. Also, as the distance from the excavations increases, the BNS stress values tend to be closer to the analytical solution. Unfortunately, stress results for all the boundary conditions, close to the excavation, deviate quite

significantly from the analytical solutions. This is to be expected due to the excavation that could not be made infinitely thin in the JFLAC models.

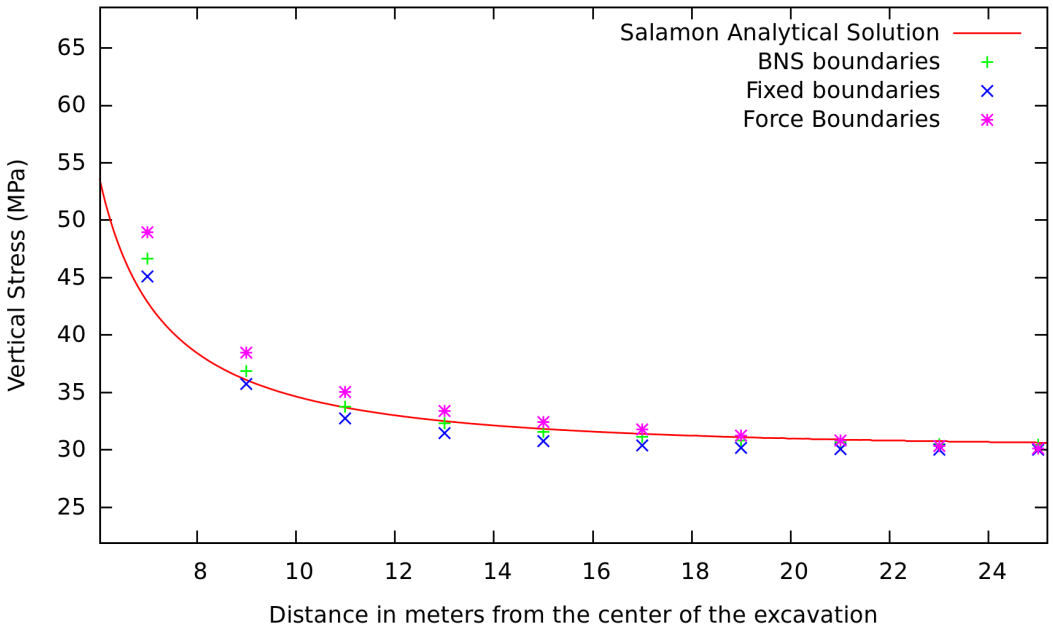


Figure 5.21: Analytical vertical stress results on a line running from an excavation, compared to solutions of the BNS-, displacement- and stress boundary conditions.

If the boundary nodes of the domain are placed close to the excavation in the above simulation, boundary effects can become more evident in the results. By using displacement- or stress boundary conditions, these effects can increase. The BNS boundary condition was developed to minimize these effects and hence make it possible to place the boundaries of the domain closer to the areas where high accuracy in the results need to be obtained. An investigation was performed to determine whether the BNS could accomplish this task.

The boundary nodes in JFLAC simulations must have either fixed displacements over the boundary plane, or a force that is calculated from a constant stress that acts on the boundary (refer to Section(3.8)). Both these boundary conditions have little effect on the accuracy of the results if they are placed at a sufficient distance from the results area. But placing boundaries far away from this area can cause the simulation model to become

large. If it is placed too close, the boundary effects become evident in the results.

Fig. (5.22) illustrates the problems that may arise when placing displacement or stress boundaries too close to an area of interest. Here a square excavation is made in an isotropic homogeneous medium under a constant stress state and the boundary nodes of the simulation model are placed close to the excavation. The closure "displacement" of the boundary is analyzed when using the displacement or stress boundary condition.

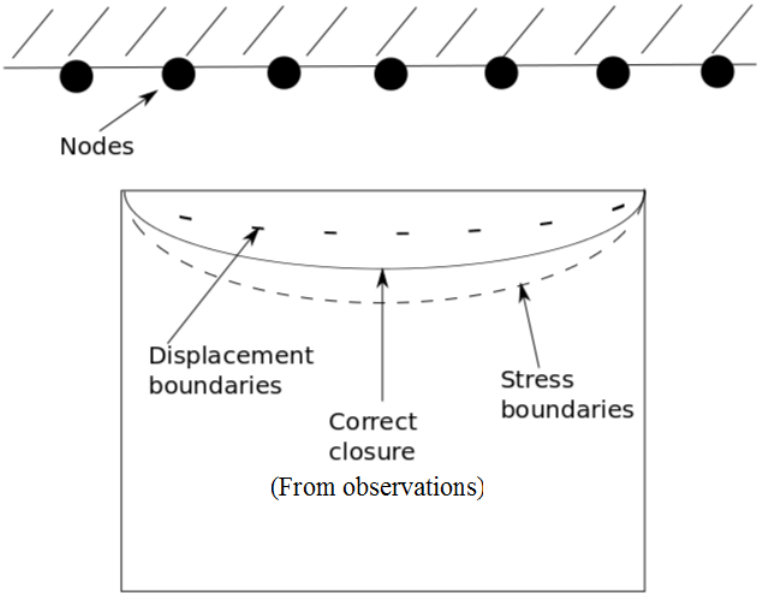


Figure 5.22: Errors in displacement and stress boundary conditions.

If the nodes on the boundary have a fixed displacement over the boundary plane, the walls of the excavation tend to show less closure than actually occurs. This is shown in Fig. (5.22). The reason for this is that the nodes on the boundary prohibit the remaining solid to move into the excavation. However, if a constant force, calculated from a constant stress that acts on the boundary, acts on the boundary nodes, more closure is measured than actually occurs. This is due to the stress on the boundary that does not decrease as the walls of the excavation move. The more the walls move, the smaller the stress in the surrounding solid becomes, which in turn should cause the stress on the boundary to

decrease.

The performance of the BNS boundary condition was compared to the displacement and fixed stress boundary conditions. A simple JFLAC model with dimensions 20 m x 20 m x 20 m in a Cartesian coordinate system was used, discritised into 1 m hexahedron zones. A 10 m x 10 m x 10 m excavation was made in the center of the model. To compare the three boundary conditions, closure on any one of the six walls of the excavation can be analyzed because of the symmetry of the problem.

Fig. (5.23) shows the closure profile that each of the three boundary conditions has on the excavation.

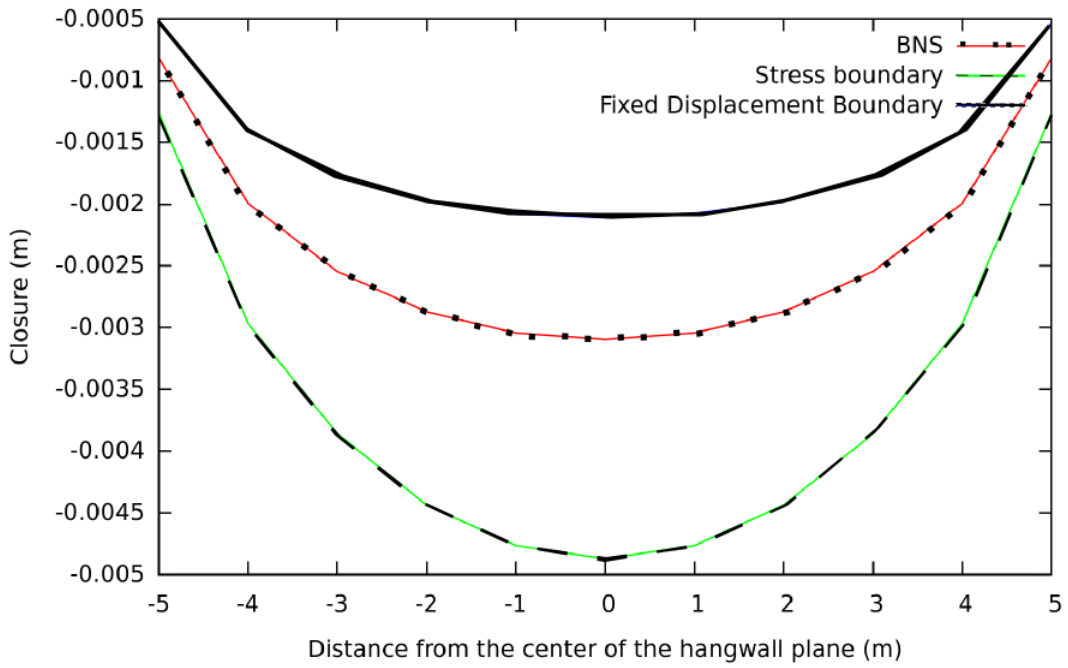


Figure 5.23: Closure profiles for the displacement, stress and BNS boundary conditions.

As expected the stress boundary condition causes the excavation walls to experience more closure than it should where the displacement boundary condition causes the ex-

cavation walls to have less closure. The BNS boundary condition allows the excavations walls to behave more realistically. This is due to the forces on the boundary nodes that are adapted to give a more realistic response as stress changes inside the domain.

If the boundaries of the domain are shifted further away from the excavation, one will find that the closure profiles of the excavation for the displacement- and stress boundary conditions tend to move towards the profile given by the BNS boundary condition. This means that if BNS boundary conditions are applied on the domain, the domain can be made smaller and computer resources can be used optimally.

Summary

The three case studies that were conducted in this section were used to test the stability, speed and accuracy of JFLAC. The first case study showed that JFLAC provided accurate results when compared to the commercial FLAC software for a well known analytical solution. This proved that JFLAC was stable and accurate enough to be used to conduct further case studies.

In the second case study JFLAC was used to perform stress analysis on a platinum mine in South Africa. The simulation models used in this study were computationally large and JFLAC solved these system in a reasonable amount of time. The stress results obtained by JFLAC produced good results and a reasonable conclusion could be made with regards to the specific problem.

The BNS boundary condition was tested and compared against the displacement- and stress boundary conditions, commonly used in FLAC. The respective boundary conditions were first compared against the results of an analytical solution. The results obtained by the BNS showed to be the closest to the results of the analytical solution. The boundary conditions were then compared by testing the convergence (closure) of a simple rectangular shaped excavation where the boundaries of the domain are placed close to the excavation.

It was found in both cases that the BNS produces more accurate results when compared to the other boundary conditions. It also allows for the boundaries of the simulated domains to be placed closer to result areas which reduces the total number of tetrahedrons in the model and saving computer resources.

Chapter 6

Summary and conclusion

6.1 Summary

The purpose of this study was to find a useful tool that can be used to asses the potential hazard of stressed rock mass in the mining industry. A stress modelling tool, FLAC, proved capable of accomplishing this task. Theoretical documentation [10] that is supplied with a copy of the FLAC software provided a good basis in understanding the mathematics involved inside the algorithm. While studying the documentation, the author of this study decided to reimplement the FLAC software in JAVA, and called it JFLAC. During the implementation the author added several improvements to the original algorithm. A new type of boundary condition, called the Boundary Node Shell (BNS) was designed to improve result accuracy of the algorithm as well as to reduce the size of the domain that was simulated. Also, since the use of multiple CPU cores in personal computers became more widely available, the author developed a multi CPU version of the algorithm which allowed for the use of more than one CPU core. This increased the performance of the algorithm significantly.

The basic governing equations used in the algorithm was covered in this document. Elasticity and Hooke's law, as well as plastic deformation, by means of the Mohr-Coulomb condition, were discussed. A detailed description of the simulated domain and it's discritisation into a Lagrangian grid was given such that the nodal formulations of the governing

equations could be derived. With all the necessary nodal equations derived, a description of the algorithm was given, providing all the necessary steps involved in implementing the algorithm. The author then described the implementation of the new BNS boundary condition and the improvements it added to the algorithm. A description of a technique for multi-threading the algorithm was also given.

Lastly, three case studies were performed to test the reliability of JFLAC. It was tested against a well-known analytical solution and was also compared against the results of FLAC for the same problem. A good comparison was found between JFLAC and FLAC and both methods showed an approximate average deviation of 2% from the analytical solution. Once the reliability was established, it was used to conduct a case study on a mine in South Africa. The results proved to be reasonable. The performance of the BNS boundary condition was then tested against the displacement- and stress boundary conditions that are implemented in FLAC. The BNS showed a significant improvement in the accuracy of the results, especially close to mining excavations.

6.2 Conclusions

JFLAC proved to be a versatile stress modelling tool that could be used to asses the potential hazard of stressed rock mass. Since it makes use of multiple CPU cores, stresses in a domain of interest can be simulated in a reasonable time frame. Highly stressed areas, or areas that failed by application of the Mohr-Coulomb condition in the domain, can be identified as possible hazardous areas in the mine where the probability of seismic events are more likely. Also, it might be used to simulate a planned mining sequence and identify future mine areas where areas can be hazardous and possibly lead to a redesign of the mined out area.

The development of the BNS boundary condition improved the accuracy of the simulated results close to mining excavations and also allowed for the reduction in the domain size, due to the elastic response of the domain boundaries.

Chapter 7

Possible improvements

One of the problems that were encountered during the study was the problem of discritising the grid. A constant mesh size was used for all the simulations. In some cases this led to unnecessary large models that reduced the performance of the algorithm. The development of a sophisticated mesh generation tool for JFLAC that discritised the modelled domain finely in areas where high accuracy in the results is needed, and more coarsely far from these areas, could reduce the total number of tetrahedrons in the domain and could improve the performance of the algorithm significantly. A graphical user interface for this tool could also reduce the amount of time needed to build the model, as this is where most of the user's time is spent.

The BNS boundary condition caused, in some cases, a large increase in the simulation time. This became more evident as the domain size increased. It is due to the large number of sources required to simulate the response of the elastic boundary. It is however possible to reduce the number of sources. This would cause the matrices of Eqs. (4.2) and (4.3) to loose their square shape and advanced decomposition techniques would be required to solve the linear system.

Lastly, for the advanced programmer it might be a nice challenge to develop a "Server Farm" multithreaded version of the algorithm.

Bibliography

[1] K.J. Bathe. *Finite Element Procedures.* Prentice Hall, 1996. 38

[2] C.A. Brebbia. *The Boundary Element Method for Engineers.* Pentech Press Limited, 1978. 3

[3] T.M. Charlton. *Energy Principles in Theory of Structures.* Oxford University Press, 1973. 53

[4] R. Courant, K. Friedrichs, and H. Lewi. On the partial difference equations of mathematic physics. *IBM Journal*, 100:32–74, 1967. 60

[5] R.O. Davis and A.P.S. Selvadurai. *Elasticity and Geomechanics.* Cambridge University Press, 1996. xii, 7, 11, 12, 13, 14, 21, 22, 46, 61

[6] R.O. Davis and A.P.S. Selvadurai. *Plasticity and Geomechanics.* Cambridge University Press, 1996. 23, 44, 45, 46

[7] W. Flugge. *Tensor Analysis and Continuum Mechanics.* Springer-Verlag Berlin Heidelberg, 1972. 9

[8] F.H. Harlow, M.A. Ellison, and J.H. Reid. The particle-in-cell computing method for fluid dynamics. *Methods Comput. Phys*, 3:319–343, 1964. 3

[9] R.E. Hunt. *Geotechnical Engineering Investigation Handbook - Second Edition.* Taylor and Francis, 2005. 12, 17, 19

[10] ITASCA. *FLAC 3 D User Guide*, 2009. 4, 24, 52, 60, 79, 106

[11] R. Madariaga. Dynamics of an expanding circular fault. *Bulletin of the Seismological Society of America*, 66:639–666, 1976. 60

[12] J. Marti and P. Cundall. Mixed discretization procedure for accurate modelling of plastic collapse. *International Journal of Numerical and Analytical Methods in Geomechanics*, 6:129–139, 1982. 49

[13] A. Mendecki and G. Van Aswegen. Seismic monitoring in mines: Selected terms and definitions. *Rockbursts and Seismicity in Mines - RaSiM5*, 5:563–570, 2001. 2

[14] A.J. Mendecki. *Seismic Monitoring In Mines*. Chapman and Hall, 1997. 2

[15] J.C. Nagtegaal, D.M. Parks, and J.R. Rice. On numerically accurate finite element solutions in the fully plastic range. *Computer methods in applied mathematics and engineering*, 4:153–177, 1974. 49

[16] C.F. Richter. An instrumental earthquake magnitude scale. *Bulletin of the Seismological Society of America*, 25:1–32, 1936. 1

[17] J.A. Ryder and A.J. Jager. *A Textbook on Rock Mechanics for Tabular Hard Rock Mines*. Creda Communications, 2002. 15, 77, 78, 79, 80, 86, 99

[18] M.D.G. Salamon. Two-dimensional treatment of problems arising from mining tabular deposits in isotropic or transversely isotropic ground. *International Journal of Rock Mechanics in Mining Science*, 5:159–185, 1968. 99

[19] J. Stewart. *Calculus - Fourth Edition*. Brook / Cole Publishing Company, 1999. 42

[20] G. Strang and G. Fix. *An Analysis of The Finite Element Method*. Prentice Hall. ISBN 0130329460, 1973. 3

[21] D. Sulsky, Z. Chen, and H.L. Schreyer. A method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118:179–196, 1994. 3

[22] R.P. Young. *Rockbursts And Seismicity In Mines*. A.A. Balkema, 1993. 2

Appendix A

Finite Difference Method

Finite difference methods are numerical approximations to the solution of differential equations by a partial derivative with a suitable algebraic difference quotient (a finite difference). By definition, the first derivative of a function $f(a)$ is given as

$$f'(a) = \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}. \tag{A.1}$$

A reasonable approximation to Eq. (A.1) would be

$$f'(a) \simeq \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}. \tag{A.2}$$

Most common finite-difference representations of derivatives are based on Taylor's series expansions and if the function can be derived for higher orders, then the Taylor series expansion for this function is

$$f(x_0 + h) = f(x_0) + \frac{f'(x_o)}{1!}h + \frac{f^{(2)}(x_0)}{2!}h^2 + ... + \frac{f^{(n)}(x_0)}{n!}h^n + R_n(x) \tag{A.3}$$

where $R_n(x)$ is the remainder term. By analyzing the first derivative of $f(a)$, then Eq. (A.3) becomes

$$f(a+h) = f(a) + f'(a)h + R_1(x) \tag{A.4}$$

and by rearranging the terms it can be shown that

$$f'(a) = \frac{f(a+h) - f(a)}{h} - \frac{R_1(x)}{h}. \tag{A.5}$$

If R_1 is sufficiently small, then is can be shown that Eq. (A.5) can be expressed as

$$f'(a) \simeq \lim_{h \to 0} \frac{f(a+h) - f(a)}{h}. \tag{A.6}$$

Only three forms of finite difference methods are commonly considered, namely the *forward difference*

$$\delta_h f(x) = f(x+h) - f(x), \tag{A.7}$$

the *backward difference*

$$\delta_h f(x) = f(x) - f(x-h), \tag{A.8}$$

and the *central difference*

$$\delta_h f(x) - f(x + \frac{1}{2}h) - f(x - \frac{1}{2}h). \tag{A.9}$$

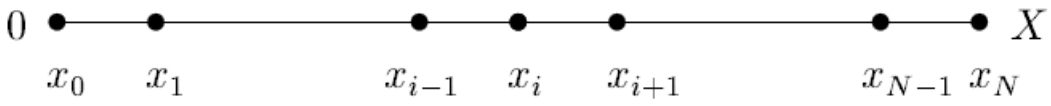


Figure A.1: An understanding of the finite difference method.

Consider grid points in 1D on a line as represented in Fig. (A.1). Each grid point can individually be represented as

$$x_i = i \triangle x \tag{A.10}$$

where

$$\triangle x = \frac{X}{N} \tag{A.11}$$

and N represents the mesh size. The first order derivative of a function $u(x)$ can be represented in terms of Eq. (A.1) as

$$\frac{\partial u(x)}{\partial x} = \lim_{h\to 0} \frac{u(x+\triangle x) - u(x)}{\triangle x}, \tag{A.12}$$

(the forward difference) or similarly as

$$\frac{\partial u(x)}{\partial x} = \lim_{h\to 0} \frac{u(x) - u(x-\triangle x)}{\triangle x} \tag{A.13}$$

(the backward difference). By adding Eq. (A.12) and Eq. (A.13) it can be shown that the central difference is

$$\frac{\partial u(x)}{\partial x} = \lim_{h\to 0} \frac{u(x+\triangle x) - u(x-\triangle x)}{2\triangle x}. \tag{A.14}$$

The Taylor series expansions for the first order derivatives of Eqs. (A.12), (A.13) and (A.14) yields

$$u_{i+1} = u_i + \triangle x \left(\frac{\partial u}{\partial x}\right)_i \tag{A.15}$$

for the forward difference and

$$u_{i-1} = u_i - \triangle x \left(\frac{\partial u}{\partial x}\right)_i \tag{A.16}$$

for the backward difference. By rearranging the terms in Eq. (A.15) and Eq. (A.16), they become

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_i}{\triangle x} \tag{A.17}$$

and

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_i - u_{i-1}}{\triangle x} \tag{A.18}$$

respectively. By adding Eqs. (A.17) and (A.18) the central difference can be expressed as

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{u_{i+1} - u_{i-1}}{2\triangle x}. \tag{A.19}$$

Appendix B

JFLAC input files

Num	Coordinate X	Coordinate Y	Coordinate Z	Boundary X	Boundary Y	Boundary Z
1	47045	-34570	-1030	1	1	1
2	47045	-34570	-1029	1	1	1
3	47045	-34570	-1028	1	1	1
4	47045	-34570	-1027	1	1	1
5	47045	-34570	-1026	1	1	1
6	47045	-34570	-1025	1	1	1
7	47045	-34570	-1024	1	1	1
8	47045	-34570	-1023	1	1	1
9	47045	-34570	-1022	1	1	1
10	47045	-34570	-1021	1	1	1
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

Num	Vertex 1	Vertex 2	Vertex 3	Vertex 4	Material	Sigma xx	Sigma yy	Sigma zz	Sigma xy	Sigma xz	Sigma yz
0	5406	5407	1	52	rock	3.052112e+07	3.052112e+07	6.104224e+07	0.000000e+00	0.000000e+00	0.000000e+00
1	5407	5408	2	53	rock	3.050544e+07	3.050544e+07	6.101088e+07	0.000000e+00	0.000000e+00	0.000000e+00
2	5408	5409	3	54	rock	3.048976e+07	3.048976e+07	6.097952e+07	0.000000e+00	0.000000e+00	0.000000e+00
3	5409	5410	4	55	rock	3.047408e+07	3.047408e+07	6.094816e+07	0.000000e+00	0.000000e+00	0.000000e+00
4	5410	5411	5	56	rock	3.045840e+07	3.045840e+07	6.091680e+07	0.000000e+00	0.000000e+00	0.000000e+00
5	5411	5412	6	57	rock	3.044272e+07	3.044272e+07	6.088544e+07	0.000000e+00	0.000000e+00	0.000000e+00
6	5412	5413	7	58	rock	3.042704e+07	3.042704e+07	6.085408e+07	0.000000e+00	0.000000e+00	0.000000e+00
7	5413	5414	8	59	rock	3.041136e+07	3.041136e+07	6.082272e+07	0.000000e+00	0.000000e+00	0.000000e+00
8	5414	5415	9	60	rock	3.039568e+07	3.039568e+07	6.079136e+07	0.000000e+00	0.000000e+00	0.000000e+00
9	5415	5416	10	61	rock	3.038000e+07	3.038000e+07	6.076000e+07	0.000000e+00	0.000000e+00	0.000000e+00
10	5416	5417	11	62	rock	3.036432e+07	3.036432e+07	6.072864e+07	0.000000e+00	0.000000e+00	0.000000e+00

Figure B.1: JFLAC model file.

```
material-name = rock
rock.density             = 2.7e3            #Material sensity
rock.shear-modulus       = 28e9             #Material shear modulus
rock.bulk-modulus        = 46e9             #Material bulk modulus
rock.youngs-modulus      = 60e9             #Material Youngs modulus
rock.friction-angle      = 30               #Material frication angle
rock.dilation-angle      = 20               #Material dialtion angle
rock.poisson-ratio       = 0.2              #Material Poisson's ration
rock.cohesion            = 5.0e6            #Material cohesion
```

Figure B.2: JFLAC material file.

```
model-name                          = Hexahedron Model 1
num-threads                         = 8                         #Number of threads to use to run simulation
velocity-damping-constant           = 0.8                       #Damping constant - alpha
large-strain-mode                   = true                      #Specify if Large Strain Model is used
flac-mode                           = hexahedron                #Input element, either hexahedrons or tetrahedrons
grid-dual-discritised               = true                      #Perform Dual Grid Discritization to zones
num-steps-before-updating-geometry  = 1                         #If the model is in Large Strain mode
material-failure-type               = ELASTIC                   #Constitutive model
nodal-mixed-discritization          = false                     #Apply Mixed Nodal Discritization to tetrahedrons
mixed-discritization                = true                      #Apply Mixed Discritization to zones
gravity                             = false                     #Use gravity in simulation
unbalanced-force-for-equilibrium    = 0.05                      #Threshold for system equilibrium
```

Figure B.3: JFLAC settings file.

In Fig. (B.1) the columns representing Vertex 1 to Vertex 4 represent the four nodes of the tetrahedron, and the integer numbers represent the specific node number in the columns above. The stress tensor as specified in Fig. (B.1) will be assigned to the tetrahedron at the simulation start and represents the virgin stress state of the tetrahedron.