Chapter 4

Numerical Simulations

In this chapter three main parts are used for the description of the sample testing. These three parts are the FE program used, the Gradient Damage code and a description of the input file.

4.1 The Finite Element Analysis Program (FEAP)

The FE program used here is a computer analysis system called FEAP [10] and it is designed for use as an instructional program to illustrate performance of different types of elements and modeling methods. This computer system includes an integrated set of modules to perform input of data describing a finite element model, construction of solution algorithms to address a wide range of applications, and graphical and numerical output of solution results. A problem solution is constructed using a command language concept in which the solution algorithm is written by the user. In the present case a Gradient Damage Code (coded by Angelo Simone, TU Delft 2000 [11]) is assigned for the definition of the materials in the sample.

4.2 Basics of the Damage Code

In order to understand properly the parameters and characteristics of the input file some explanations have to be made about the code used. For this, some preliminaries are introduced related to Continuum damage mechanics and a justification of its use in this project.

4.2.1 Softening and Localisation

Many of the engineering materials show a mechanical behaviour that is composed by a linear elastic branch until the tensile strength is reached, followed by a decrease of the material stiffness. In this last stage we can distinguish a hardening behaviour until the peak load is reached and a softening behaviour from the peak until a residual strength is reached or the complete failure of the sample is achieved.

In the micro-structural level it is possible to explain the softening behaviour: multiple micro-cracks will be responsible for the material damage. These micro-cracks propagate through the ITZ and the matrix of the material forming a failure band. Strain will tend to concentrate in the failure band and the rest of the material will tend to unload. This strain concentration and localization will enable the global structure to reproduce the softening behaviour mentioned previously. When considering an elasticity based damage constitutive relation in a quasi-brittle material the main dissipative processes can be related to the degradation of the elastic constitutive moduli [11].

4.2.2 Basics of Continuum Damage Mechanics

Damage mechanics is a branch of continuum mechanics that incorporates changes in the micro-structural level via a finite number of scalar or tensor-valued internal variables (Lemaitre and Chabosche 1990 [12]). Damage is related to plasticity because the material history on the stress evolution is also incorporated in the continuum theory [13], although it differs from plasticity because in a damage context unloading does not occur elastically. This is caused by the degradation of the elastic stiffness matrix considered in the unloading branch. Generalizing to three dimensions and assuming that all damage effects can be reproduced by a single variable we can relate the stress tensor, σ, and strain tensor, ϵ, through

$$\sigma = (1-\omega)\mathbf{D^e}\varepsilon, \tag{4.1}$$

ω being an internal variable (damage parameter) which signifies how much of the system is damaged. $\omega = 0$ denotes a virgin material with no damage while $\omega = 1$ denotes a fully damaged material. $\mathbf{D^e}$ is the elastic stiffness matrix which can be expressed in terms of the Lamé constants or both Young's modulus E and the Poisson's's ratio ν. Damage growth is controlled by the damage loading function

$$f(\tilde{\varepsilon}, k) = \tilde{\varepsilon} - k, \tag{4.2}$$

where k is a history dependent parameter which reflects the loading history and $\tilde{\varepsilon}$ an equivalent strain that is an invariant of the strain tensor. The history parameter goes from a initial value k_0 where the damage is supposed to start and grows memorizing the larger value of the $\tilde{\varepsilon}$. In the present study a Mazar's definition is considered for the $\tilde{\varepsilon}$ as a function of principal strains

$$\tilde{\varepsilon} = \sqrt{\sum_{i=1}^{3}(<\varepsilon_i>)^2}, \tag{4.3}$$

where $<\varepsilon_i>=\varepsilon_i$ if $\varepsilon_i > 0$ and $<\varepsilon_i>=0$ otherwise. The history parameter k never decreases so its rate $\dot{k}$ is strictly positive. The structure of the loading function can, just as for plasticity, be formalized using the Kuhn-Tucker conditions

$$f \leq 0,\ \dot{k} \geq 0,\ f\dot{k} = 0. \tag{4.4}$$

Furthermore, an evolution law for the damage variable ω is defined as a function of the history parameter $\omega = \omega(k)$.

In our case an exponential softening damage evolution law is chosen. For this the following relation holds:

$$\omega = 1 - \frac{k_0}{k}[1 - \alpha + \alpha e^{-\beta(k-k_0)}], \tag{4.5}$$

where α and β are material parameters: β controls the slope of the softening branch (giving more negative slopes for larger values of β) and α sets the residual strength.

4.2.3 Gradient-Enhanced Damage Formulation

It is possible to translate a non-local quantity, given by an averaging integral relation such as

$$\bar{q}(x) = \frac{\int_\Omega g(\xi) q(x+\xi) d\Omega(\xi)}{\int_\Omega g(\xi) d\Omega(\xi)}, \tag{4.6}$$

into an equivalent differential formulation using only the computation of derivatives proposed in [11]. Considering $\tilde{\varepsilon}$ as state variable, this differential formulation can be expressed in the spirit of

$$\tilde{\varepsilon}(x) = \tilde{\varepsilon}_{nl}(x) - c\nabla^2 \tilde{\varepsilon}_{nl}(x), \tag{4.7}$$

$\tilde{\varepsilon}_{nl}(x)$ being the non-local equivalent strain and $\tilde{\varepsilon}(x)$ the local equivalent strain. Moreover $c = \frac{l^2}{2}$ and l is the internal length scale of the gradient enhancement. This parameter means the internal length scale needed to regularise[1] the localisation of deformation and is related to the width of the localisation band.

The natural boundary condition specified for the previous equation is

$$\mathbf{n}^T \nabla \tilde{\varepsilon}_{nl} = 0. \tag{4.8}$$

In the previous chapter it is mentioned that small changes in the mesh size for ITZ, matrix and particle phases were needed in order to be able to construct the samples with a reasonable number of elements and avoiding that two or more particles can remain in contact. These little changes in the length of the elements do not affect the results, because the element length remain small enough for any case to fit into the smallest failure band given for the length scale.

Mathematical formulation

The Governing equations are both equilibrium in the domain and implicit gradient-enhancement

$$\begin{aligned} \mathbf{L}^T \sigma + \mathbf{b} &= \mathbf{0} \\ \tilde{\varepsilon}(x) &= \tilde{\varepsilon}_{nl}(x) - c\nabla^2 \tilde{\varepsilon}_{nl}(x). \end{aligned} \tag{4.9}$$

The Boundary Conditions are the ones defined on Γ_t, Γ_u and the natural boundary conditions on $\partial\Omega$

$$\begin{aligned} \mathbf{N}^T \sigma &= \hat{\mathbf{t}} \\ \mathbf{u} &= \hat{\mathbf{u}} \\ \mathbf{n}^T \nabla \tilde{\varepsilon}_{nl} &= 0. \end{aligned} \tag{4.10}$$

[1] We talk about regularisation as a way to avoid that the governing equations change type locally.

The Constitutive Relations specify (in this order) the stress/strain relation, the damage evolution law, the history variable definition, the equivalent strain definition, the loading function and the Kuhn-Tucker loading unloading conditions

$$\begin{gathered}\sigma = (1-\omega)\mathbf{D^e}\varepsilon \\ \omega = \omega(k) \\ k = max(k; \tilde{\varepsilon}_{nl}(\tau), \tau \leq t) \\ \tilde{\varepsilon}_{nl} = \tilde{\varepsilon}_{nl}(\varepsilon) \\ f(\tilde{\varepsilon}_{nl}) = \tilde{\varepsilon}_{nl} - k\tilde{\varepsilon}_{nl} \\ f \leq 0, \dot{k} \geq 0, f\dot{k} = 0,\end{gathered} \tag{4.11}$$

where $\mathbf{L}$ is defined as

$$\mathbf{L^T} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \end{bmatrix}. \tag{4.12}$$

Stress and strain tensors are represented as follows:

$$\sigma^{\mathbf{T}} = \begin{bmatrix} \sigma_{xx} & \sigma_{yy} & \sigma_{zz} & \sigma_{xy} & \sigma_{yz} & \sigma_{zx} \end{bmatrix} \tag{4.13}$$

$$\varepsilon^{\mathbf{T}} = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{yy} & \varepsilon_{zz} & 2\varepsilon_{xy} & 2\varepsilon_{yz} & 2\varepsilon_{zx} \end{bmatrix}. \tag{4.14}$$

Stresses and strains satisfy the boundary conditions (Eqs. (4.10)) where $\mathbf{N}$ is related to the normal vector $\mathbf{n}$ by

$$\mathbf{N^T} = \begin{bmatrix} n_x & 0 & 0 & n_y & 0 & n_x \\ 0 & n_y & 0 & n_x & n_z & 0 \\ 0 & 0 & n_z & 0 & n_y & n_x \end{bmatrix}. \tag{4.15}$$

Finite Element Method Implementation

If we multiply the equilibrium equations by a virtual displacement, using Green's formula and considering the boundary conditions we obtain the weak form of the equilibrium equations:

$$\int_\Omega \delta\varepsilon^{\mathbf{T}}\sigma \mathbf{d\Omega} = \int_{\mathbf{\Omega}} \delta\mathbf{u^T b d\Omega} + \int_{\mathbf{\Gamma_t}} \delta\mathbf{u^T \hat{t} d\Gamma}. \tag{4.16}$$

In a similar fashion we can process Eq. (4.7) and obtain:

$$\int_\Omega \delta\tilde{\varepsilon}_{nl}\tilde{\varepsilon}_{nl}d\Omega + \int_\Omega \nabla\delta\tilde{\varepsilon}_{nl}^T c \nabla\tilde{\varepsilon}_{nl}d\Omega = \int_\Omega \delta\tilde{\varepsilon}_{nl}\tilde{\varepsilon}d\Omega. \tag{4.17}$$

For the spatial discretisation of the weak form the variables $\tilde{\varepsilon}_{nl}$ and $\mathbf{u}$ are discretised using standard finite element shape functions. The two-field discretisation, in each element can be written as

$$\mathbf{u} = \mathbf{H_u u} \tag{4.18}$$

$$\tilde{\varepsilon}_{nl} = \mathbf{H}_\varepsilon \tilde{\boldsymbol{\varepsilon}}_{\mathbf{nl}}, \tag{4.19}$$

where $\mathbf{H_u}$ and $\mathbf{H}_\varepsilon$ are matrices containing the shape functions,

$$\mathbf{H_u} = \begin{bmatrix} h_{u_1} & 0 & 0 & h_{u_2} & 0 & 0 & \cdots & h_{u_n} & 0 & 0 \\ 0 & h_{u_1} & 0 & 0 & h_{u_2} & 0 & \cdots & 0 & h_{u_n} & 0 \\ 0 & 0 & h_{u_1} & 0 & 0 & h_{u_2} & \cdots & 0 & 0 & h_{u_n} \end{bmatrix} \tag{4.20}$$

$$\mathbf{H}_\varepsilon = \begin{bmatrix} h_{\varepsilon_1} & h_{\varepsilon_2} & \cdots & h_{\varepsilon_n} \end{bmatrix}. \tag{4.21}$$

By substitution of the discretized quantities in the weak formulation the discretized form of the field equations is obtained,

$$\int_\Omega \mathbf{B_u^T} \sigma \mathbf{d\Omega} = \int_{\mathbf{\Omega}} \mathbf{H_u^T b d\Omega} + \int_{\mathbf{\Gamma_t}} \mathbf{H_u^T \hat{t} d\Gamma} \tag{4.22}$$

$$\int_\Omega \mathbf{H_\varepsilon^T H}_\varepsilon \tilde{\varepsilon}_{\mathbf{nl}} + \mathbf{B_\varepsilon^T c B}_\varepsilon \tilde{\varepsilon}_{\mathbf{nl}} d\Omega = \int_\Omega \mathbf{H_\varepsilon^T} \tilde{\varepsilon} \mathbf{d\Omega}, \tag{4.23}$$

where $\mathbf{B_u} = \mathbf{L H_u}$ and $\mathbf{B}_\varepsilon = \nabla \mathbf{H}_\varepsilon$

To obtain a pseudo-time discretisation we introduce the linearisation at iteration i with respect to the previous iteration. Then, at a nodal level the discretisation can be written as

$$u_i = u_{i-1} + \Delta u_i, \tag{4.24}$$

$$\tilde{\varepsilon}_{nl,i} = \tilde{\varepsilon}_{nl,i-1} + \Delta \tilde{\varepsilon}_{nl,i}. \tag{4.25}$$

Substituting in the discretized weak form we finally obtain

$$\begin{bmatrix} \mathbf{K^{uu}_{i-1}} & \mathbf{K^{u\varepsilon}_{i-1}} \\ \mathbf{K^{\varepsilon u}_{i-1}} & \mathbf{K^{\varepsilon\varepsilon}_{i-1}} \end{bmatrix} \begin{bmatrix} \mathbf{\Delta u_i} \\ \Delta \tilde{\varepsilon}_{nl,i} \end{bmatrix} = \begin{bmatrix} \mathbf{\Delta f^u_i} \\ \mathbf{\Delta f^\varepsilon_i} \end{bmatrix} = \begin{bmatrix} \mathbf{\Delta f^u_{ext}} \\ 0 \end{bmatrix} - \begin{bmatrix} \mathbf{f^u_{int,i-1}} \\ \mathbf{f^\varepsilon_{int,i-1}} \end{bmatrix}, \tag{4.26}$$

where the sub-matrix $\mathbf{K}^{()}_{()}$ are expressed as integrals of $\mathbf{B}, \mathbf{D}$ and $\mathbf{H}$. The internal force vectors are defined as follows:

$$\mathbf{f^u_{ext}} = \int_{\mathbf{\Omega}} \mathbf{H_u^T b d\Omega} + \int_{\mathbf{\Gamma_t}} \mathbf{H_u^T \hat{t} d\Gamma} \tag{4.27}$$

$$\mathbf{f^u_{int,i-1}} = \int_{\mathbf{\Omega}} \mathbf{B_u^T} \sigma_{\mathbf{i-1}} \mathbf{d\Omega} \tag{4.28}$$

$$\mathbf{f^\varepsilon_{int,i-1}} = \int_{\mathbf{\Omega}} \mathbf{H_\varepsilon^T H}_\varepsilon \tilde{\varepsilon}_{\mathbf{nl,i-1}} + \mathbf{B_\varepsilon^T c B}_\varepsilon \tilde{\varepsilon}_{\mathbf{nl,i-1}} - \mathbf{H_\varepsilon^T} \tilde{\varepsilon}_{\mathbf{nl,i-1}} \mathbf{d\Omega}. \tag{4.29}$$

The final system of equations represented in Eq. (4.26) is clearly non-linear as many variables are function of the displacements $\mathbf{\Delta u}$. For the solution of the nonlinear system an iterative method is used (Full Newton-Raphson in our case). The components of the stiffness matrix are updated in every displacement step allowing to solve displacements, strains, stresses and the force vectors once a certain convergence criteria is achieved.

Thus, an algorithm for a Gradient-Enhanced Damage Model can be written as:

1. Update the stiffness matrix $\mathbf{K}^{()}_{()}$
2. Solve for Δu_i and $\Delta \tilde{\varepsilon}_{nl,i}$ (according to Eq. 4.26)

3. Update u_i and $\tilde{\varepsilon}_{nl,i}$ at the nodal points
4. In the integration points:
   - Compute the strain increment $\Delta\varepsilon_{\mathbf{i}}$
   - Update the total strain $\varepsilon_{\mathbf{i}} = \varepsilon_{\mathbf{i-1}} + \mathbf{\Delta}\varepsilon_{\mathbf{i}}$
   - Update the total non-local equivalent strain $\tilde{\varepsilon}_{\mathbf{nl,i}} = \tilde{\varepsilon}_{\mathbf{nl,i-1}} + \mathbf{\Delta}\tilde{\varepsilon}_{\mathbf{nl,i}}$
   - Compute the equivalent strain $\tilde{\varepsilon}_i = \tilde{\varepsilon}(\varepsilon_{\mathbf{i}})$
   - Evaluate the loading function $f = \tilde{\varepsilon}_{nl,i} - k_{i-1}$ and check for loading
   - if loading ($f > 0$) then $k_i = \tilde{\varepsilon}_{nl,i}$ else $k_i = k^a_{i-1}$
   - Update the Damage variable $\omega_i = \omega(k_i)$
   - Compute the new stress $\sigma_i = (1 - \omega_i)\mathbf{D^{el}}\varepsilon_{\mathbf{i}}$
5. Update the internal forces (Eqs. (4.28, 4.29))
6. Check for Convergence

4.3 Description of the Input File

In this section a description of the parts forming the input file is made including details, remarks and difficulties appeared during its generation.

4.3.1 Mesh Inclusion

The first part of the input is used to include the file, obtained by NEGE, containing the elements and node coordinates forming the mesh. On the top of the file we can see a number of parameters that will characterize our problem. These are the number of nodal points, the number of elements, the number of material property sets (3 in our case), the space dimension of the mesh (2D for this study), maximum number of degrees of freedom per node (3 DOFs in this case[2]), maximum number of nodes per element (3 as we are working with 3-noded triangles).

4.3.2 Boundary Conditions

For this study a tension test is made by fixing one of the sample boundaries and imposing a certain displacement to the opposite one, as it can be seen in Figure 4.1.

The prescribed displacement is imposed in 200 displacement steps with a length of 1.10^{-5} length units. This particular displacement step is chosen in order to allow all the samples to converge properly in every step. Special care has to be paid when selecting the length of the displacement step using a damage code. We should work with a small enough displacement step in order to make sure that we obtain a quadratic convergence in each one. In this particular study, tests are made with smaller displacement step to verify that the correct solution is obtained.

[2]Working with gradient damage, we have 2 unknown spatial variables (x,y components of the displacements) and the equivalent strain

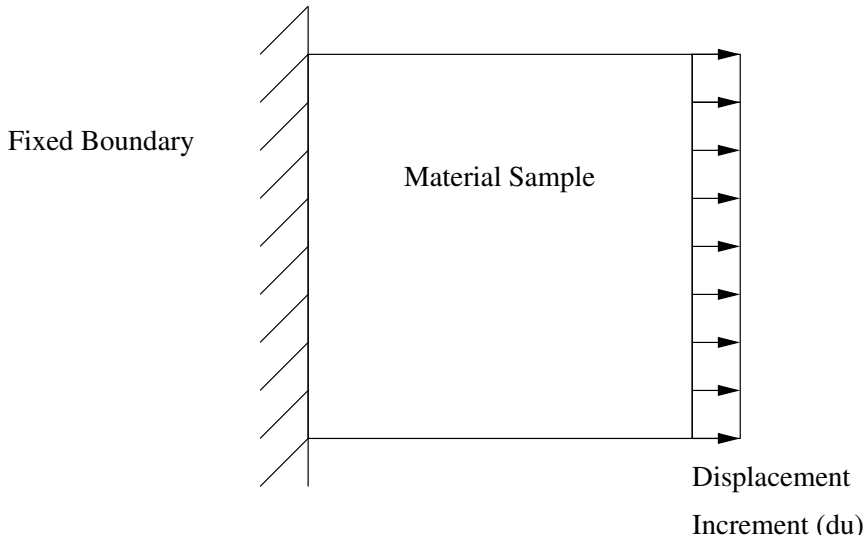

Figure 4.1: Illustration of the boundary conditions

4.3.3 Material properties and damage parameters

In the next part the material properties are specified for the matrix, particles and ITZ.

In the first place we find information about the code used (in our case is the gradient damage code). After that, it is specified that we work with isotropic elasticity in the elastic branch where the values of the Young's modulus and Poisson's ratio are introduced. In our case we chose the same value for the Poisson's ratio in both three materials and a different Young's modulus between matrix, particles and ITZ. The particles are stiffer than the matrix and the matrix is also thought to be stiffer than the ITZ.

It is also specified that we work with plane strain. This means that there is no value for the strain in the third dimension so the strain tensor has only the components of the x-y plane. We chose a value of 0.2 (squared unit lengths) for the c parameter of the damage model. This means that the internal length scale l is equal to 0.63 (length units) approximately. Bearing in mind that the length of the biggest element in the sample that can be in a damage area is 0.38 (this is the biggest element length for the matrix phase), the damaged band can be clearly reproduced by at least one of this biggest elements without regularisation problems.

It is important to remark that, in this study, c is considered as a material parameter and for that, a similar width of the damage band is assumed for all samples. Other studies relate the thickness of shear bands with the microstructure of the material. It is the case of experiments in sands (Vardoulakis and Graf 1985 [14]), but the relation between the thickness of the band and the size of the unit cell sample is not clear.

The parameters α and β have been chosen in order to be able to view clearly the whole behaviour of the material within these 200 displacement steps. Therefore we

found $\alpha = 0.9$ and $\beta = 1500$.

The value of k_0 is different for each material. In order to reproduce with accuracy what is observed in the reality, damaged is imposed to begin in the ITZ. After that the matrix will enter in damage as well but the particles are not supposed to be damaged during the whole test. For this, the smallest value of k_0 is assigned to the ITZ, the matrix will have a larger k_0 and finally the particles will have such a large value of k_0 that damage will not take place because the deformation rates reached during the test will be much smaller. Therefore, we can make sure that the particles will behave elastically during the whole deformation path. The mentioned values are: $k_0 = 3.10^{-6}$ for the ITZ, $k_0 = 5.10^{-6}$ for the matrix and finally an artificially large value $k_0 = 0.5$ is set for the particles.

4.3.4 Computation Block

The last part of the input file contains two loops. A first loop is created to update the displacement step. Inside this loop, another one is used to calculate the stiffness matrix with a certain maximum number of iterations until the tolerance is reached. This tolerance is set to 1.10^{-6} and the maximum number of iterations in each step is set to 10.

Next, some instructions are given in order to obtain the outputs and plots expected such as the "Force vs Displacement" graphic. This force is an average of the reactions acting in the right boundary (Figure 4.1). For each value of the displacement imposed in the right boundary we obtain its corresponding force average.

Other interesting plots are obtained, such as images of the distribution of a certain variable in the sample cell for a particular displacement step. The plots selected show:

- The three phases composing the material and the reactions in the boundaries.
- The distribution of the damage in the sample cell.
- The distribution of the stress in the direction of the displacement imposed (x-direction).
- The distribution of the equivalent strain in the damage sample.

The following box contains the input file described previously.

```
FEAP

2834, 5464, 3, 2, 3, 3

INCL,30101.msh

EBOUN

1 -5.153 1 1
1 5.153 1 1

EDISP

1 -5.153 0.0 0.0
1 5.153 1.0 0.0

MATErial,1!,MATRIX
USER 03
ELASTIC ISOTROPIC 25000.0 0.2
PLANe STRA
QUAD DATA 2 2
UCON GRAD 0.2
UCON GDDA 2 0.000005d0 0.9d0 1500.d0
UCON GDEQ 3

MATErial,2!PARTICLES
USER 03
ELASTIC ISOTROPIC 30000.0 0.2
PLANe STRA
QUAD DATA 2 2
UCON GRAD 0.2
UCON GDDA 2 0.5d0 0.9d0 1500.d0
UCON GDEQ 3

MATErial,3!ITZ
USER 03
ELASTIC ISOTROPIC 20000.0 0.2
PLANe STRA
QUAD DATA 2 2
UCON GRAD 0.2
UCON GDDA 2 0.000003d0 0.9d0 1500.d0
UCON GDEQ 3
END

tie

BATC
OPTI,ON
TOL,,1.d-6
PROP,,1
DT,,0.00001
PDEL,fu30101.x
LOOP,time,200
TIME
LOOP,iter,10
UTANG
FORM
SOLV
next,iter
PLOT,WIPE
plot,fram,1
PLOT,fill
plot,bound
plot,react
plot,fram,2
plot,stre,5
PLOT,FRAM,3
PLOT,STRE,1
PLOT,FRAM,4
PLOT,CONT,3
PLOT,DEFO,1,1
PDEL,fu30101.x
next,TIME

END

2,2
0.0,0.0 0.002,0.002

INTE

STOP
```