Fall 11-28-2016

High Performance Computing Applied to Structural Analysis of Concrete Structures

Jose A. Mena Ortiz
University of New Mexico

Jose Abraham Mena Ortiz
Candidate

Civil Engineering
Department

This thesis is approved, and it is acceptable in quality and form for publication:

Approved by the Thesis Committee:

Dr. Walter Gerstle , Chairperson

Dr. Mahmoud Reda Taha

Dr. Susan Atlas

Dr. Patrick Bridges

HIGH PERFORMANCE COMPUTING APPLIED TO STRUCTURAL ANALYSIS OF CONCRETE STRUCTURES

by

Jose Abraham Mena Ortiz

B.S., Civil Engineering, University of Granada, Spain, 2015

THESIS

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Civil Engineering

The University of New Mexico

Albuquerque, New Mexico

December, 2016

Dedication

To my parents, Pepi and Jose, my brother, Ruben, and my grandmother, Francisca, for their unconditional support and love.

Acknowledgments

I want to thank my advisor and committee chair Dr. Gerstle for his support, guidance and patience throughout this research. Thank you for the opportunity to work with you. Thank you Dr. Bridges for helping me on the parallel processing part of this thesis. I would also like to thank Dr. Taha and Dr. Atlas for agreeing to serve on my committee.

I want to thank the Center for Advanced Research Computing for the computational resources used in this work.

I want to thank my parents, Pepi and Jose, my brother, Ruben, my grandmother, Francisca, and my uncle, Vicente, for their encouragement. I want to also thank my colleagues Jose, Serafin, Sherif and Siavash for their help and support.

HIGH PERFORMANCE COMPUTING APPLIED TO STRUCTURAL ANALYSIS OF CONCRETE STRUCTURES

by

Jose Abraham Mena Ortiz

B.S., Civil Engineering, University of Granada, Spain, 2015

MS., Civil Engineering, University of New Mexico, 2016

Abstract

Continuum mechanics models are the main tool in structural analysis. Due to their continuity assumption, some of them do not predict accurately the behavior of concrete structures. While these models are widely used by design engineers, many are flawed for fundamental reasons.

The State-Based Peridynamic Lattice Model (SPLM) is presented in this thesis as a viable alternative to continuum models. The SPLM is shown to have a simple formulation that allows the engineer to fully understand the underlying theory. The elastic, plastic and damage SPLM models are presented. Moreover, SPLM is shown to be capable of modelling essential mechanisms in concrete structures.

SPLM is run on massive parallel computers, since it requires large computational power. In this thesis, a new parallel implementation is presented, and a study of the performance of the original and the new system is presented.

Contents

List of Figures

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Over the past 200 years, continuum mechanics has been the main tool in structural analysis. However, the classical theory of continuum mechanics has been demonstrated to have difficulties modelling concrete structures. These models are based on the assumption of continuity. Over the years, continuum models have been developed to capture the inelastic behavior of concrete. Consequently, continuum models have become more and more complex, while they are widely accepted and used. We formulate the questions: Why not instead of continuing this trend, do we not go to the origin of the problem? Why don't we discard the basic assumption of continuity, and start building a more realistic and simple model from the foundations?

Peridynamics is a non local solid modelling theory that has the capability of modelling a body as a finite number of particles. Continuum peridynamics has been developed since 2000, by Silling (Silling, 2000), with the purpose of solving the fundamental problems of continuum models. Gerstle (Gerstle, 2015) developed the

State-Based Peridynamic Lattice Model (SPLM). SPLM has been demonstrated to be capable of reasonably modelling concrete structures, but still presents some issues and limitations.

The motivation for this thesis is to answer the previous questions, and to continue working on SPLM, studying the basis of the model, detecting its problems and limitations, and to possibly overcome them.

1.2 Scope of Thesis

The objective of this thesis is to present an alternative to classical continuum models. Continuum models, while used by most of the design engineers, are difficult to understand and to use, and do not realistically predict the response of concrete structures. We want to develop a simpler model that would be easily understandable by a design engineer, and will predict sufficiently accurately the behaviour of concrete structures. We choose SPLM for this purpose. But if we want SPLM to successfully complement continuum models in design engineering, we must further develop SPLM models.

SPLM requires significant computational power, which can be a limitation. In this thesis, we introduce a new parallel implementation. We also study and identify some performance issues, with the aim of blazing the path for future improvement.

1.3 Outline of Thesis

This thesis includes seven chapters: Introduction, Literature Review, New Peridynamic Damage Models, New SPLM Plasticity Model, SPLM Reinforced Concrete Beams, High Performance Computing, and Conclusions.

Chapter two presents a literature review that provides the reader with the

background necessary to understand this thesis. This chapter reviews some classical models in fracture, damage and plasticity, and the simplified modified compression field theory, and its application in the Canadian Code (CSA A23.3- 04, 2004).

Chapter three introduces, the elastic SPLM developed by Gerstle, the Bond-based Peridynamic Lattice Damage Model (BPLDM), and presents a study on lattice rotation. In this chapter we also present the development of the new SPLM damage model, that is justified by the results of the BPLDM.

Chapter four focuses on the new SPLM plasticity model. This chapter also includes a comparison of the new SPLM models with the classical continuum and Abaqus models (Simulia, 2015).

Chapter five presents the application of these new SPLM models to analysis of reinforced concrete beams. This chapter also includes a new bond-slip model and a study of the size effect.

Chapter six provides some basics on high performance computing, the implementation of a parallel implementation, and a scalability analysis of the original and new implementation.

Finally, chapter seven includes a brief summary, major findings and suggestions for future research in SPLM.

Chapter 2

Literature Review

Concrete is one of the most common materials used in structures such as buildings, bridges and dams. Concrete presents the problem that nonlinear cracking behavior occurs well before structural failure. Crack initiation and propagation is one of the most important aspects in the failure analysis of concrete structures (Lee and Fenves, 1998). Many authors have worked on developing fracture, damage and plasticity models that simulate the real response of concrete structures. In this literature review we analyze some of the classical and state-of-the-art damage and plasticity models. Finite Element Analysis (FEA) models are also presented using Abaqus (Simulia, 2015). Finally, the Simplified Modified Compression Field Theory (Bentz, et al., 2006) and the Canadian Code for concrete structures (CSA-A23.3-04, 2004) are presented, since their equations are used as a comparison with the SPLM for prediction of the shear strength of reinforced concrete beams.

2.1 Damage And Plasticity Models in Concrete

The short-term deformation modes in concrete are essentially elastic deformation, tensile cracking and compressive crushing. Tensile cracking has been widely modelled by applying fracture and damage mechanics models, such as the fictitious crack model (Hillerborg et al., 1976; Hillerborg et al., 1978). Compressive crushing has been usually modelled using the classical theory of plasticity. In recent years, researchers have developed continuum constitutive models that couple these two damage modes (Lubliner et al., 1989; Lee and Fenves, 1998).

2.1.1 Fictitious Crack Theory

The fictitious crack model was developed by Hillerborg (Hillerborg et al., 1976; Hillerborg, 1978). The basic idea is to model a "fictitious" crack that can transfer stress across a discrete crack until a certain point where the material stiffness degradation is complete. The model is based on the discrete fracture mechanics approach (Bazant and Planas, 1998).

The fracture zone is assumed to start at any point of the specimen where the first principal stress reaches the tensile strength of the material. The development of this fracture zone is perpendicular to the first principal stress direction. The cohesive closing traction is assumed to be a function of the crack opening displacement (COD or w), as shown in Figure 2.1. Outside the fracture process zone the material is assumed to follow linear elastic constitutive relations.

According to classical fracture mechanics, in the process of developing a unit area of crack surface, a certain amount of energy, G_f is absorbed. A crack will propagate when the released energy (G) is equal of greater than the absorbed energy (G_f). Thus, the amount of energy absorbed per unit crack area in opening the crack

from 0 to its limit (w_1) is :

$$\int_0^{w_1} \sigma dw = G_f \tag{2.1}$$

which is the area under the stress vs. crack opening displacement (w) curve, as shown in Figure 2.1.

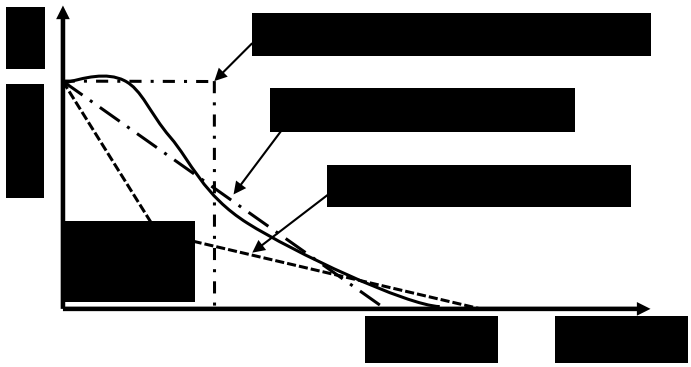


Figure 2.1: Variation of stress σ with crack opening displacement w.

To fit the $\sigma - w$ curve some authors have proposed the use of a constant function (Dugdale, 1960), a linear function (Hillerborg et al., 1978), a bilinear function (Hillerborg et al., 1976; Wittmann et al., 1988) or an exponential function. Any approximation must fulfill the essential condition that the area under the curve must be equal to the energy absorbed, G_f.

The fictitious crack model is beautiful in its simplicity, but it is perhaps over simplified. Since the stiffness degradation function is based on the absorbed energy required to open a tensile unit crack area, it is completely insensitive to the triaxial stress conditions.

2.1.2 Plastic Damage Models for Concrete

Lubliner and coauthors (Lubliner et al., 1989) proposed a constitutive model based on an internal variable formulation. The internal scalar variables are the dam-

age variable, the elastic degradation variable and the plastic degradation variable. These parameters are fracture-energy-based and they are calibrated using experimental results.

Lubliner's model is based on the Mohr-Coulomb and Drucker-Prager yield criteria, which have the form:

$$F(\boldsymbol{\sigma}) = c \tag{2.2}$$

where $F(\boldsymbol{\sigma})$ is a function of the stress components and c may be identified with the cohesion. A plastic-damage variable κ, that replaces the classical "hardening variable" is introduced. κ never decreases and increases if and only if plastic deformation takes place. With this new variable, c is scaled to be $c = f_{co}$, the initial yield strength in uniaxial compression, when $\kappa = 0$ and $c = 0$ when $\kappa = 1$. Nonetheless, c is not necessary a function of κ. Instead, c is an internal variable governed by its rate equation $\dot{c}$, which is proportional to $\dot{\kappa}$.

The introduction of these internal variables modify the classical equations for damage and plasticity. For simplicity, the governing equations of the model are presented next. Equation 2.2 represents the yield criterion, Equation 2.3 is the elastic-plastic strain decomposition, Equation 2.4 is the flow rule, Equation 2.5 represents the rate equation for κ and Equation 2.6 is the rate equation for c.

$$\boldsymbol{\varepsilon} = \boldsymbol{\varepsilon}^e + \boldsymbol{\varepsilon}^p = \mathbf{D}^{-1}\boldsymbol{\sigma} + \boldsymbol{\varepsilon}^p \tag{2.3}$$

$$\dot{\varepsilon}^p = \dot{\boldsymbol{\lambda}}\mathbf{g} \tag{2.4}$$

$$\dot{\kappa} = h^T(\boldsymbol{\sigma}, c, \kappa)\dot{\boldsymbol{\varepsilon}}^p \tag{2.5}$$

$$\dot{c} = \kappa(\boldsymbol{\sigma}, c, \kappa)\dot{\kappa} \tag{2.6}$$

where $\mathbf{D}$ is the undamaged stiffness tensor, $\boldsymbol{\sigma}$ is the stress tensor, $\dot{\lambda}$ is the plastic loading factor, $\mathbf{g} = \partial G/\partial\boldsymbol{\sigma}$ is the plastic flow vector normal to the plastic potential

surface $G = const.$ and h^T is the viscoplastic potential function.

The governing equations of this model can be challenging for a practicing engineer to understand since they require a considerable background in formal continuum mechanics and plasticity theory. In addition, it is not obvious how the internal variables represent physical effects. However, let us continue presenting the model, in order to have a global perspective.

This model requires that κ is computed using a fracture mechanics energy approach. It is necessary to have a $\sigma_c - \varepsilon^p$ curve to compute the portion of κ that accounts for compression, and a $\sigma_t - \varepsilon^p$ curve to compute the tension portion. This approach presents a fundamental problem. When the material is under tension and a crack initiates, the strain is not homogeneous in the structure since the crack creates a discontinuity, that makes the strain to jump theoretically to infinity. Therefore, the model fails by relying on the plastic strain, which is not applicable near the crack tip, and it depends upon the localization and specimen size. Moreover, these curves are impossible to obtain objectively, since it is unclear how to get the plastic strain part out of the total displacement experimentally.

Following the derivation presented in (Lubliner et al., 1989), Equation 2.7 contains the yield surface (see Figure 2.2).

$$F(\sigma) = \frac{1}{1-\alpha}\left[\sqrt{3J_2} + \alpha I_1 + \beta\langle\sigma_{max}\rangle - \gamma\langle-\sigma_{max}\rangle\right], \tag{2.7}$$

where I_1 and J_2 are the first and second stress invariants, respectively, and

$$\begin{aligned}
\alpha =& \frac{(f_{b0}/f_{c0}) - 1}{2(f_{b0}/f_{c0}) - 1}, \\
\beta =& (1-\alpha)(f_{c0}/f_{t0}) - (1+\alpha), \\
\gamma =& \frac{3(1-\rho)}{\rho - 1}, \\
\rho =& \frac{\gamma + 3}{2\gamma + 3}.
\end{aligned}$$

Lee and Fenves (Lee and Fenves, 1998) modified Lubliner's model. The damage

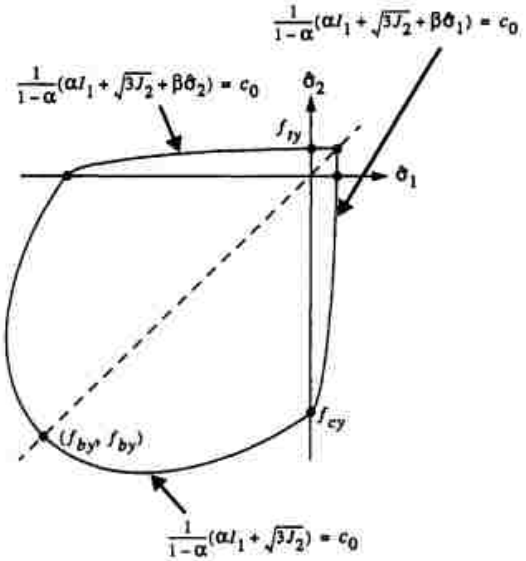


Figure 2.2: Lubliner yield surface in plane stress space. From (Lee and Fenves, 1998).

states are represented by two variables, one for tensile damage (κ_t) and the other one for compressive damage (κ_c). Furthermore, the yield function is modified by using multiple damage variables.

Lubliner's yield function is based on the Mohr-Coulomb and Drucker-Prager function (Equation 2.2), with some variations. The yield function depends on the stress ($\boldsymbol{\sigma}$) and two state variables, f_t and f_c, which are the uniaxial tensile and compressive strength of the material respectively:

$$\tilde{F}(\sigma, f_t, t_c) \leq 0. \tag{2.8}$$

The uniaxial strength functions can be expressed as functions of the effective stress responses $\bar{f}_t$ and $\bar{f}_c$:

$$f_t = [1 - D_t(\kappa_t)]\bar{f}_t(\kappa_t), \tag{2.9}$$

$$f_c = [1 - D_c(\kappa_c)]\bar{f}_c(\kappa_c), \tag{2.10}$$

where D_c and D_t are the tensile and compressive damage function respectively.

The plastic strain rate $\dot{\boldsymbol{\varepsilon}}^p$ and rate equation for κ are given by the following

equations:

$$\dot{\boldsymbol{\varepsilon}}^p = \dot{\lambda}\left(\frac{s}{\|s\|} + \alpha_p \mathbf{I}\right), \tag{2.11}$$

$$\dot{\kappa} = \dot{\lambda}\hat{\mathbf{H}}(\hat{\bar{\sigma}}, \kappa), \tag{2.12}$$

where $\dot{\lambda}$ is the plastic consistency parameter, $\mathbf{s}$ is the deviatoric stress, α_p is the coefficient of plastic potential function, $\mathbf{I}$ is the first stress invariant and $\hat{\bar{\sigma}}$ is the eigenvalue matrix for effective stress tensor.

Finally, the yield surface from (Lubliner et al., 1989) has been modified by introducing the damage variable κ. Thus, the expression for the yield surface becomes:

$$F(\sigma, \kappa) = \frac{1}{1-\alpha}\left[\alpha I_1 + \sqrt{3J_2} + \beta(\kappa)\langle\hat{\sigma}_{max}\rangle\right] - c_c(\kappa), \tag{2.13}$$

where α and β are coefficients of the yield function and c_c is the compressive degradation damage variable.

These two models may replicate accurately the real behavior of concrete, but they are complicated and difficult to fully understand. Lubliner and coauthors (Lubliner et al, 1989) proposed a constitutive model that using the classical theory of plasticity can couple the tensile and compressive damage and plasticity. Their approach was to add a set of internal variables to the plasticity equations, in order to model damage effects. This approach is interesting, but it makes the model extremely complicated. In addition, the plastic-damage parameter κ is obtained using a combination of $\sigma_c - \varepsilon^p$ and $\sigma_t - \varepsilon^t$ curves. As stated before, by definition, "strain" is not applicable when there is a discontinuity, and the strain is dependent on how it is measured and the localization size. In addition, if we neglect the definition of strain, how to experimentally extract the plastic strain from the total strain is not clear, and since, the plastic strain is rate dependent is not uniquely defined. Lee and Fenves (Lee and Fenves, 1998) tried to solve this problem by modifying Lubliner's model, introducing two damage variables (κ_t and κ_c) instead of one. This modification slightly simplified the model, but again, it is based on a fundamental assumption

that I find unreasonable.

2.2 Abaqus Models

Abaqus is one of the most advanced commercial finite element programs. The reason for reviewing the Abaqus models is to use them as a comparison with the peridynamic models developed in this thesis.

In this section we analyse the most important aspects related to this thesis, of the three Abaqus models for concrete, from the simplest to the most complex.

2.2.1 Abaqus Smeared Crack Model

The Smeared Crack Model is based on Smeared Cracking theory (Bazant and Planas, 1998) to describe the post-cracking response of both plain and reinforced concrete. According to the Abaqus User's Guide (Simulia, 2015), the model is intended to simulate concrete behavior for essentially monotonic loadings under fairly low confining pressures.

Cracking initiates when the stresses reach the "crack detection surface", shown in Figure 2.3, which is basically a linear relationship between the hydrostatic pressure stress, **p**, and the Mises deviatoric stress, **q**. Additional cracking at the same point is restricted to be orthogonal to the direction of the principal stress. The model does not track individual "macro" cracks. Instead, constitutive calculations are performed independently at each integration point of the finite element model.

The post-peak response of concrete is modelled using a tension stiffening relation. The user can specify the tension stiffening function by a stress-strain relation or a fracture energy cracking criterion.

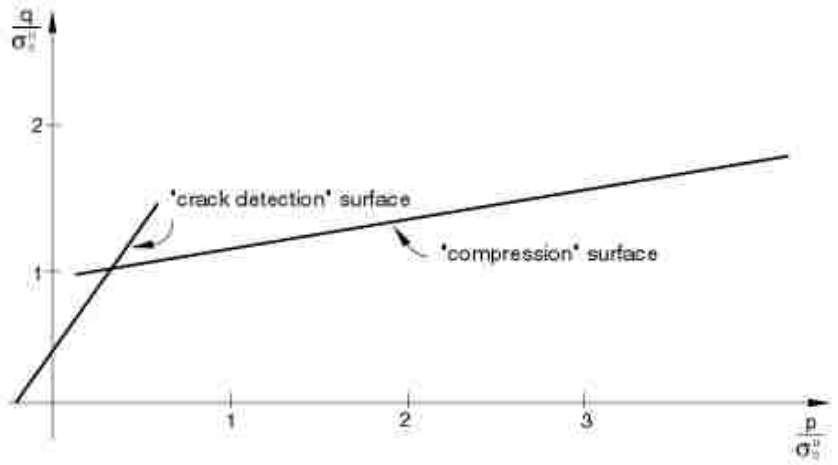


Figure 2.3: Yield and failure surfaces in the (p-q) plane. From (Simulia, 2015).

The stress-strain relations are defined by the parameters that describe the curve (Figure 2.4). The curve shape depends on factors such as the density of the reinforcement, the quality of the bond between the rebar and the concrete, the relative size of the concrete aggregate compared to the rebar diameter, and the finite element mesh. This curve should be redefined, since after the failure point, the "strain" is dependent on the mesh size. Therefore, every time the mesh is refined, the curve changes.

The fracture energy cracking criterion makes use of Hillerborg's fictitious crack model (Hillerborg et al., 1976). The post-failure response is characterized by a stress-displacement curve, as shown in Figure 2.5. This approach requires the definition of a characteristic length associated with an integration point, to ensure solution stability, and a displacement, u_0, at which the stress decreases to zero. This ultimate displacement is determined from the fracture energy per unit area definition (Equation 2.14). This approach also presents the inconvenience that it is mesh sensitive. The ultimate displacement is given by:

$$u_0 = \frac{2G_f}{\sigma_t^u}, \tag{2.14}$$

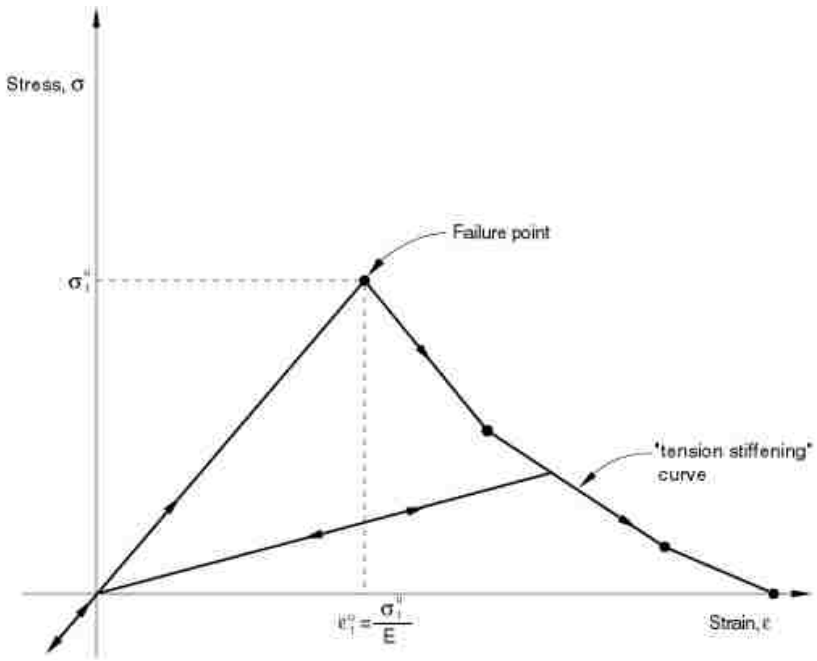


Figure 2.4: Tension stiffening model. From (Simulia, 2015).

where σ_t^u is the maximum tensile stress that the concrete can carry.

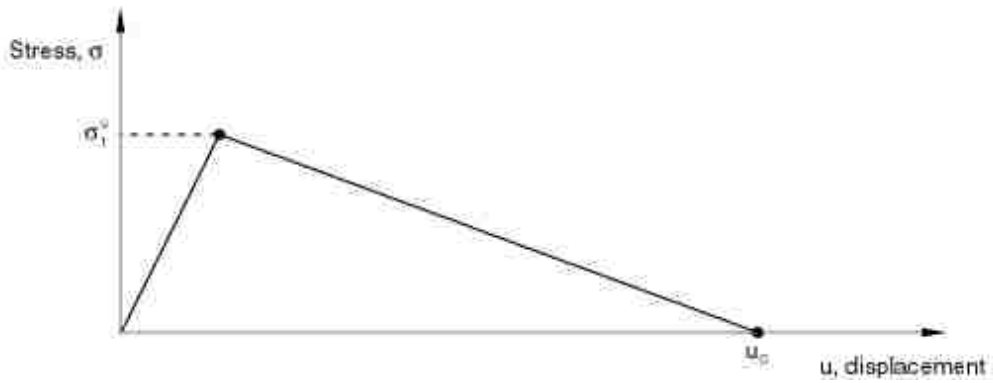


Figure 2.5: Fracture energy cracking model. From (Simulia, 2015).

In this model, the compressive behavior of concrete follows the classical theory of plasticity, with the Mises yield surface, an associated flow and an isotropic hardening function. For plane stress conditions, the yield surface simplifies as illustrated

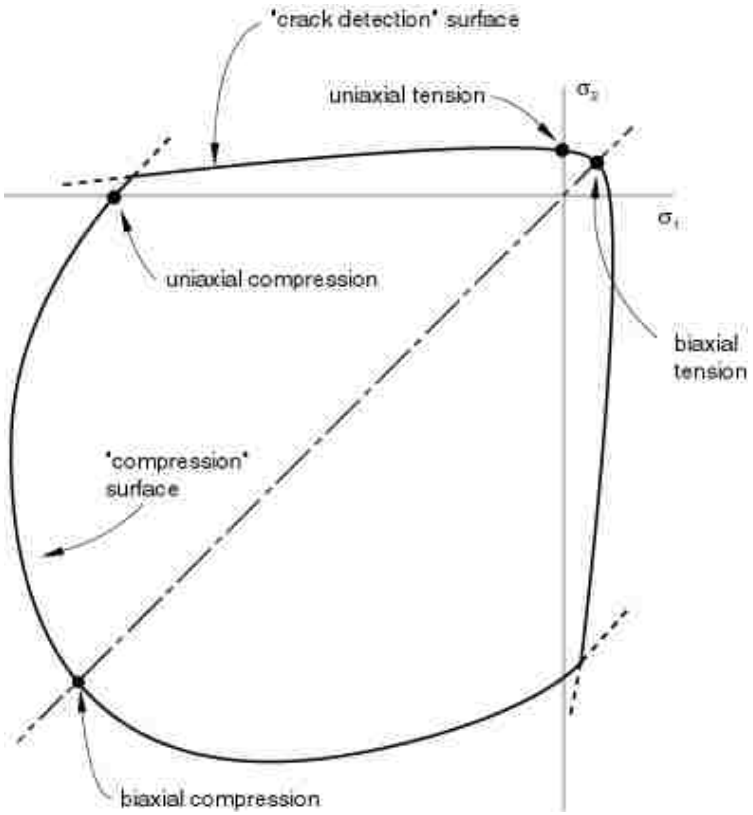


Figure 2.6: Yield and failure surfaces in plane stress conditions. From (Simulia, 2015).

in Figure 2.6.

This model is reasonable for modelling tensile cracking damage and problems where plasticity or compressive damage do not govern. Its plasticity model is over-simplified because it does not take into account excessive plastic flow as a damage mode. In addition, as with most finite element models, it is mesh sensitive.

2.2.2 Abaqus Brittle Cracking Model

The brittle cracking model is intended for modeling specimens in which the tensile cracking is the dominant mode of failure. It models the compressive behaviour

always under linear elastic conditions. In addition, it assumes no plastic response.

The cracking initiation and evolution conditions are the same as in the smeared crack model. There are some differences from the smeared crack model regarding the definition of the post-failure tension stiffening. Thus, the shape of the post-failure cracking strain vs. stress is different, as shown in Figure 2.7. Likewise, the fracture

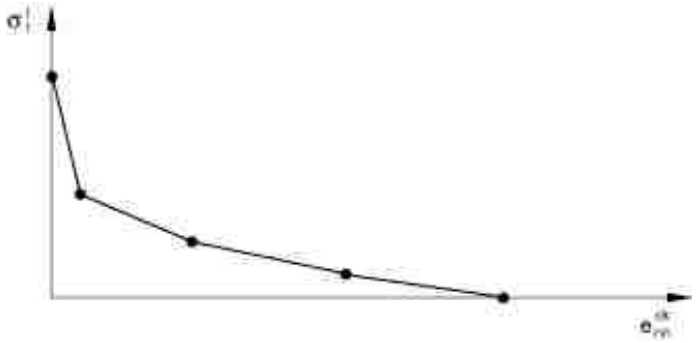


Figure 2.7: Post-failure stress-strain curve. From (Simulia, 2015).

energy cracking criterion follows Hillerborg's fictitious crack model.

This model, like the fictitious crack model, is simple and more appropriate to model tensile cracking than the smeared crack model. Nevertheless, it is not complete since it does not include plasticity and damage due to excessive plastic flow.

2.2.3 Abaqus Concrete Damaged Plasticity Model

The last Abaqus concrete model is the most complex and complete. The concrete damaged plasticity model is a continuum, plasticity-based, damage model for concrete. In the smeared crack and brittle cracking models the primary failure mechanism was tensile cracking. In this model compressive crushing is added as well. As described in the User's manual (Simulia, 2015), the inelastic and damage behaviour of concrete is represented as a combination of isotropic damaged elasticity in conjuction with non-associated multi-hardening plasticity.

The degradation of the elastic stiffness (d_t and d_c) and the evolution of the yield surface are characterised, similar to Lee and Fenves (Lee and Fenves, 1998), by:

$$\begin{aligned} d_t &= d_t(\tilde{\varepsilon}_t^{pl}, \theta, f_i), \quad 0 \leq d_t \leq 1 \\ d_c &= d_c(\tilde{\varepsilon}_c^{pl}, \theta, f_i), \quad 0 \leq d_c \leq 1 \end{aligned} \tag{2.15}$$

where, $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$ are the tensile and compressive equivalent plastic strains, θ is the temperature, and f_i represents other predefined field variables.

The "effective" tensile and compressive stress are defined in Equation 2.16 as a function of the initial (undamaged) elastic stiffness E_0, and the equivalent plastic strains:

$$\begin{aligned} \bar{\boldsymbol{\sigma}}_t &= \frac{\sigma_t}{(1-d_t)} = \mathbf{E}_0(\boldsymbol{\varepsilon}_t - \tilde{\boldsymbol{\varepsilon}}_t^{pl}), \text{and} \\ \bar{\boldsymbol{\sigma}}_c &= \frac{\sigma_c}{(1-d_c)} = \mathbf{E}_0(\boldsymbol{\varepsilon}_c - \tilde{\boldsymbol{\varepsilon}}_c^{pl}). \end{aligned} \tag{2.16}$$

Similar to the other two models, the tension stiffening response can be defined using a stress-strain relation or a fracture energy cracking criterion. The fracture energy criterion, is again based on the Hillerborg's fictitious crack model approach to compute the strain softening curve.

To define the post-failure stress-strain relations, the post-failure stress is given as a function of the cracking strain, $\tilde{\varepsilon}_t^{ck}$. Figure 2.8 presents graphically the definition of cracking strain in the stress-strain curve. Again, this model presents the problem of being mesh sensitive. The cracking strain values are converted to plastic strains using the following equation:

$$\tilde{\boldsymbol{\varepsilon}}_t^{pl} = \tilde{\boldsymbol{\varepsilon}}_t^{ck} - \frac{d_t}{(1-d_t)} \frac{\boldsymbol{\sigma}_t}{E_0} \tag{2.17}$$

Analogous to the approach to define the tension stiffening, the compressive behavior is described using a stress-strain relation. The hardening effect is described in terms of inelastic strain, $\tilde{\varepsilon}_c^{in}$, instead of plastic strain, $\tilde{\varepsilon}_c^{pl}$. The compressive inelastic

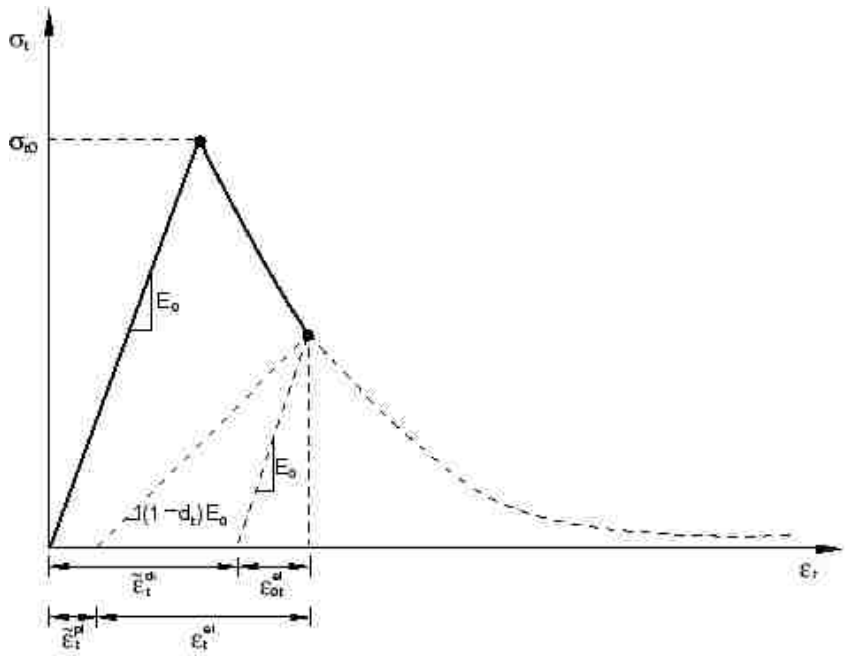


Figure 2.8: Illustration of the definition of the cracking strain $\tilde{\varepsilon}_t^{ck}$

strain is defined in Equation 2.18. Figure 2.9 shows the hardening and softening response in compression.

$$\tilde{\varepsilon}_c^{in} = \varepsilon_c - \varepsilon_{0c}^{el}; \quad \varepsilon_{0c}^{el} = \frac{\sigma_c}{E_0} \tag{2.18}$$

The plastic flow potential function and the yield surface make use of the two stress invariants: hydrostatic stress ($\bar{p}$) and the Mises effective stress deviator ($\bar{q}$). The plastic flow is assumed to be nonassociated. The Drucker - Prager hyperbolic function (Lee and Fenves, 1988) is used to compute the plastic potential function G:

$$G = \sqrt{(\varepsilon\sigma_{to}\tan(\psi))^2 + \bar{q}^2} - \bar{p}\tan(\psi), \tag{2.19}$$

where $\psi(\theta, f_i)$ is the dilatation angle measured in the $p-q$ plane at high confining pressure, $\sigma_{to}(\theta, f_i)$ is the uniaxial stress at failure, and $\varepsilon(\theta, f_i)$ is a parameter referred to as the eccentricity, that defines the rate at which the function approaches the

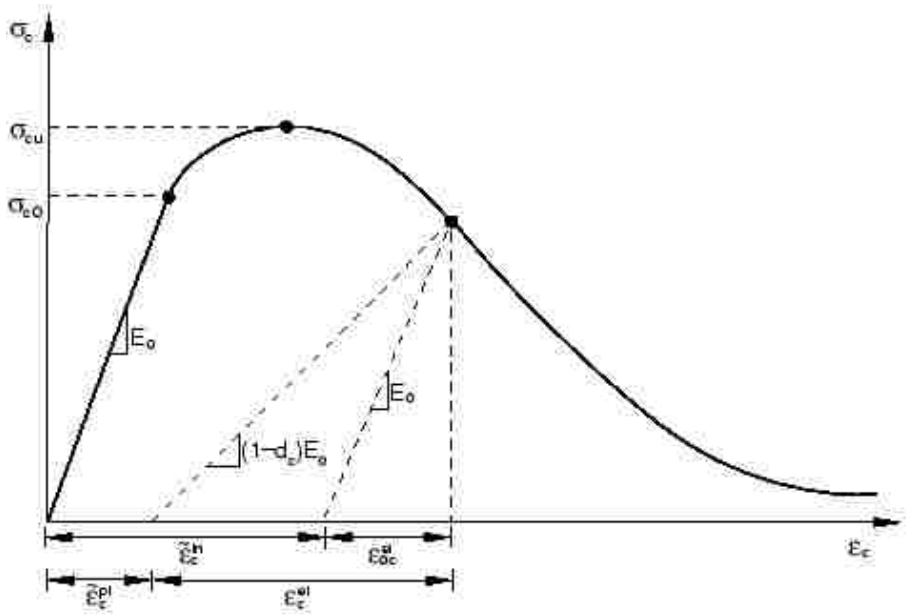


Figure 2.9: Definition of the compressive inelastic strain $\tilde{\varepsilon}_c^{in}$ used for the definition of compression hardening data. From (Simulia, 2015).

asymptote.

The yield function utilised, was developed by (Lee and Fenves, 1998), who modified the yield function from (Lubliner et al., 1989). The evolution of the yield surface is controlled by the hardening variables $\tilde{\varepsilon}_t^{pl}$ and $\tilde{\varepsilon}_c^{pl}$, that account for different evolution of damage under tension and compression.

In Equation 2.20 the yield function is expressed in terms of the effective stresses.

$$F = \frac{1}{1-\alpha}\Big(\bar{q} - 3\alpha\bar{p} + \beta(\tilde{\varepsilon}^{pl})\langle\bar{\sigma}_{max}\rangle - \gamma\langle\tilde{\sigma}_{max}\rangle\Big) - \bar{\sigma}_c(\tilde{\varepsilon}_c^{pl}) = 0 \tag{2.20}$$

with

$$\alpha = \frac{(\sigma_{bo}/\sigma_{co})}{2(\sigma_{bo}/\sigma_{co}) - 1}, \quad 0 \leq \alpha \leq 0.5, \tag{2.21}$$

$$\beta = \frac{\bar{\sigma}_c(\tilde{\varepsilon}_c^{pl})}{\bar{\sigma}_t(\tilde{\varepsilon}_t^{pl})}(1-\alpha) - (1+\alpha), \tag{2.22}$$

and

$$\gamma = \frac{3(1 - K_c)}{2K_c - 1}. \tag{2.23}$$

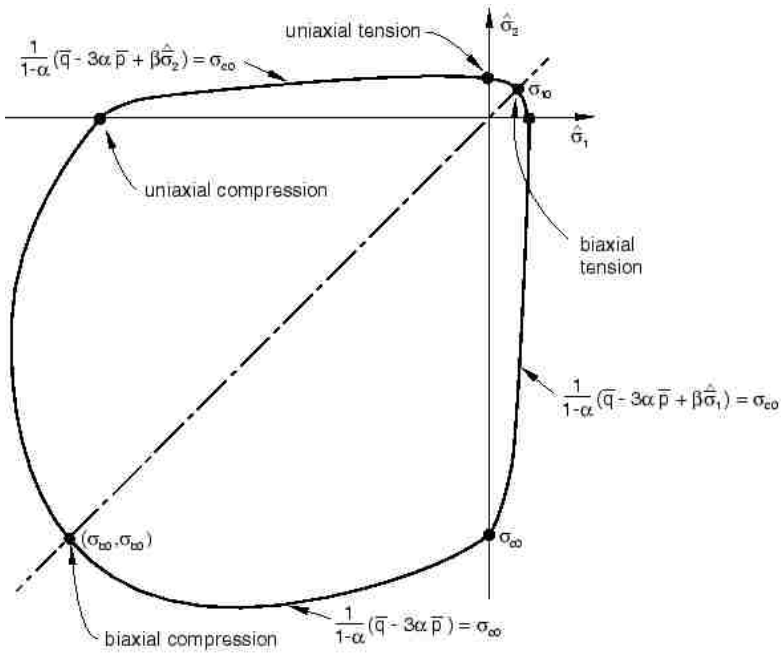


Figure 2.10: Yield surface in plane stress. From (Simulia, 2015).

This model is extremely complex and also difficult to replicate due to the large number of parameters used. In addition, some of the parameters non-physical, i.e. they are coefficients whose only purpose is to fit the experimental data. Because the model is based upon strain, which becomes infinite as damage evolves, it is non objective.

2.3 Simplified Modified Compression Field Theory and CSA-23.3-04

The modified compression field theory (MCFT) is an analytical model for analysis of load-deformation response of reinforced concrete beams undergoing normal and in-plane shear stresses. The MCFT was developed by Vecchio and Collins (Vecchio and Collins, 1986). It assumes a series of evenly distributed cracks. Similar to a cohesive crack model, these cracks can transmit tensile stress between its faces. Furthermore, they are oriented normal to the principal strain direction, as shown in Figure 2.11.

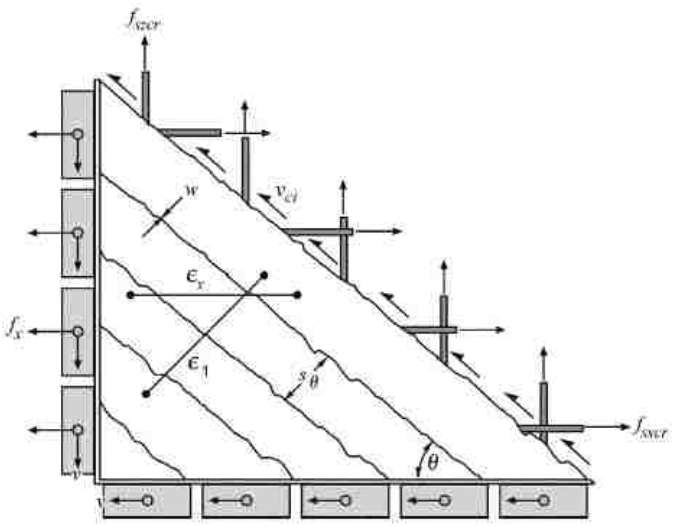


Figure 2.11: Transmission of forces across cracks. From (Bentz et al., 2006).

The first problem that arises when applying this theory is that the coefficients are based on experimental results of reinforced beams with specific sizes and steel ratios. The model gives a reasonable prediction for structures that have been used to calibrate the equations, but may fail to predict problems with different geometry or mechanical properties. The second issue, is that for membrane elements it is

difficult to find the relationships between the axial and shear stresses, as well as the axial and shear strains. For this last reason, Bentz, Vecchio and Collins (Bentz et al., 2006) modified the MCFT, adding some simplifications, and developed the Simplified Modified Compression Field Theory (SMCFT).

The SMCFT assumes shear stress failure, when failure occurs before yielding of the transverse reinforcement, to be $0.25f'_c$, where f'_c is the uniaxial compressive strength. If failure occurs below this shear stress level, it assumes that the failure at both f_{sz} (stress in the shear reinforcement) and f_{szcr} (stress in the shear reinforcement at the crack location) are equal to the yield stress of the transverse reinforcement, f_y.

The shear strength of the structure is the summation of the concrete shear strength plus the steel reinforcement:

$$v = v_c + v_s = \beta\sqrt{f'_c} + \rho_z f_y \cot\theta, \tag{2.24}$$

where ρ_z is the transverse shear ratio and θ is the inclination of the crack (see Figure 2.11). In elements with shear reinforcement β is:

$$\beta \leq \frac{0.18}{0.31 + 24w(a_g + 16)}. \tag{2.25}$$

The crack width w is computed as the product of the crack spacing s_θ and the principal tensile strain ε_1. a_g is the maximum coarse aggregate size in mm. The cracking spacing s_θ can be simplified to s_x, which is the vertical distance between bars in the x-direction.

For elements with no shear reinforcement, $s_\theta = s_x/\sin\theta$, and β is expressed as:

$$\beta \leq \frac{0.18}{0.31 + 0.686 s_{xe}\varepsilon_1/\sin\theta}, \tag{2.26}$$

where

$$s_{xe} = \frac{35 s_x}{a_g + 16}. \tag{2.27}$$

When the values of ε_x and s_{xe} are very small, β can be rewritten as:

$$\beta = \frac{0.4}{1 + 1500\varepsilon_x} \cdot \frac{1300}{1000 + s_{xe}}. \tag{2.28}$$

Finally, the angle of inclination θ can be computed as:

$$\theta = (29^\circ + 7000\varepsilon_x)\left(0.88 + \frac{s_{xe}}{2500}\right) \leq 75^\circ. \tag{2.29}$$

The Canadian *CSA A23.3 - 0.4 - Design of Concrete Structures* (CSA A23.3-04) makes use of the SMCFT equations with some modifications for shear prediction. Section 11.3, *Design for shear and torsion in flexural members* contains the equations for computing the shear strength in reinforced concrete beams.
The factored shear resistance, V_r is calculated as:

$$V_r = V_c + V_s + V_p, \tag{2.30}$$

where V_s is the shear strength of the reinforcement and V_p is the shear strength of the prestress reinforcement. V_c is computed as follows:

$$V_c = \phi_c \lambda \beta \sqrt{f'_t} b_w d_v, \tag{2.31}$$

where ϕ_c is a concrete resistance factor, λ is a factor to account for low-density concrete, b_w is the beam web width or minimum effective width, and d_v is the effective shear depth, taken as the greater of $0.9d$ or $0.72h$. β can be computed using the *Simplified Method* or the *General Method.*

The simplified method can be used when the yield strength of the longitudinal reinforcement does not exceed 400 MPa and the concrete strength does not exceed 60 MPa. θ is simplified to be 35° and β is computed as follows:

(a) If the section contains at least the minimum transverse reinforcement $\beta = 0.18$.

(b) If the section contains no transverse reinforcement and $a_g > 20mm$: $\beta = \frac{230}{1000+d_v}s$

(c) For the same conditions than the previous one but for all aggregate sizes, d_v is replaced by

$$s_{ze} = \frac{35 s_z}{15 + a_g} \tag{2.32}$$

However, s_{ze} is taken as less than $0.85 s_z$. The crack spacing parameter, s_z, is taken as d_v or as the maximum distance between layers of distributed longitudinal reinforcement, whichever is less.

In the general method β is calculated as:

$$\beta = \frac{0.4}{1 + 1500\varepsilon_x} \cdot \frac{1300}{1000 + s_{ze}}. \tag{2.33}$$

For sections containing at least the minimum transverse reinforcement $s_{ze} = 300mm$. Otherwise, s_{ze} is computed using Equation 2.32. If f'_c exceeds 70 MPa, the term a_g is taken as zero in Equation 2.32. As f'_c goes from 60 to 70 MPa, a_g is linearly reduced to zero.

The angle of inclination, θ, of the diagonal compressive stresses is calculated as:

$$\theta = 29 + 7000\varepsilon_x. \tag{2.34}$$

The longitudinal strain, ε_x, at mid-depth of the cross-section is computed from Equation 2.35, which accounts for the shear resistance that the longitudinal reinforcement provides.

$$\varepsilon_x = \frac{M_f/d_v + V_f - V_p + 0.5N_f - A_p f_{po}}{2(E_s A_s + E_p A_p)}. \tag{2.35}$$

In evaluating Equation 2.35, the following conditions apply:

(a) V_f and M_f is taken as positive and $M_f \geq (V_f - V_p)d_v$.

(b) In calculating A_s , the area of bars that are terminated less than their development length from the section under consideration are reduced in proportion to their lack of full development.

(c) If the value of ε_x calculated from Equation 2.35 is negative, it is taken as zero or the value shall be recalculated with the denominator of Equation 2.35 replaced by $2(E_s A_s + E_p A_p + E_c A_{ct})$. However, ε_x is not taken as less than $-0.20 \cdot 10^{-3}$.

(d) For sections closer than d_v to the face of the support, the value of ε_x calculated at d_v from the face of the support may be used in evaluating β and θ.

(e) If the axial tension is large enough to crack the flexural compression face of the section, the resulting increase in ε_x is taken into account. In lieu of more accurate calculations, the value calculated from Equation 2.35 is doubled.

(f) β and θ may be determined from Equations 2.33 and 2.34, respectively, using a value of ε_x that is greater than that calculated from Equation 2.35. However, ε_x is not taken greater than $3.0 \cdot 10^{-3}$.

The MCFT and the SMCFT are two analytical models that predict very accurately the shear strength of reinforced concrete beams. However, some of the basic assumptions of the model non-physical, like the homogenous distribution of cracks parallel to the normal direction of the principal stresses or the ability of a crack to transfer stresses. Furthermore, I think that it is a big assumption the principle that the aggregate interlock is one of the mechanisms that allows the cracks to transfer tensile stress. However, we have to recognise that the model predicts accurately the response of "standard" reinforced concrete beams, and captures the size effect reasonably, but it is necessary to know its limitations.

In this chapter some continuum plastic damage models have been presented. Abaqus FEA models have been chosen to demonstrate the complexity and flaws of the continuum models. The MCFT and SMCFT have been introduced as analytical models to analyse concrete beams. In the next chapter the elastic and damage SPLM models are introduced as a simpler alternative to model short terms inelastic response of concrete.

Chapter 3

New Peridynamic Damage Models

In the literature of finite element models (Lubliner et al., 1989; Lee and Fenves, 1998), tensile and compressive damage, and plasticity are coupled phenomena. Consequently, these models are complex since the constitutive equations incorporate these phenomena. In SPLM tensile damage and compressive damage are two uncoupled modes, providing much simpler and intuitive models.

In this chapter the SPLM elasticity model for plane stress conditions is developed. Two new peridynamic damage models for tensile cracking are presented. The first is the Bond Peridynamic Lattice Damage Model (BPLDM), which is an anisotropic model because each bond in a particle neighbourhood can take an independent value of damage. The BPLDM is shown to be sensitive to lattice rotation; hence, it is not general enough to accurately model the basic concrete tests (uniaxial tension, uniaxial compression and Brazilian split cylinder). The State Based Peridynamic Lattice Damage Model is next introduced and tested. Its development is justified by the need for the BPLDM to be more general and accurate.

3.1 Elastic SPLM

In 2000 Silling published a paper called *Reformulation of Elasticity Theory for Discontinuities and Long-Range Forces* (Silling, 2000). In this work he proposed the original continuum peridynamics formulation. The term peridynamic comes from the Greek roots 'peri' (near) and 'dynamic' (force). A pairwise force function between each pair of particles in peridynamics is the equivalent constitutive relationship in continuum mechanics. This pairwise force is a function only of the relative position and relative displacement between the pair of particles. That is the reason why this model is also called bond-based peridynamic model.

The bond-based peridynamic model was found to be not general enough to model solid mechanics problems. In 2007 Silling presented a more general formulation, the state-based peridynamic model (Silling et al., 2007). In this model the pairwise force function is not only a function of the relative displacement and position of a pair of particles, but also of the positions of other particles inside its material horizon, δ.

In Silling's formulation a body is treated as a continuum in $\mathbb{R}^3$. Gerstle introduced the State-based Peridynamic Lattice Model (SPLM) (Gerstle, 2015), that assumes a particle lattice rather than a particle continuum.

The linear elastic SPLM has the capability of approximately simulating linear (small deformation) and non-linear (large deformation) elastic classical continuum problems. In this section we are presenting the framework for the elastic SPLM in plane stress conditions.

Figure 3.1 shows the lattice topology in 2D as well as the order of the particles. This configuration presents the advantage that the particles can be easily stored in an array. Every particle has a neighbour list ($\mathcal{N}[P_i]\langle B_j\rangle$), which are the particles inside the peridynamic horizon (δ). Thus, at each time step each particle "knows" the particles it has to interact with, saving computation time.

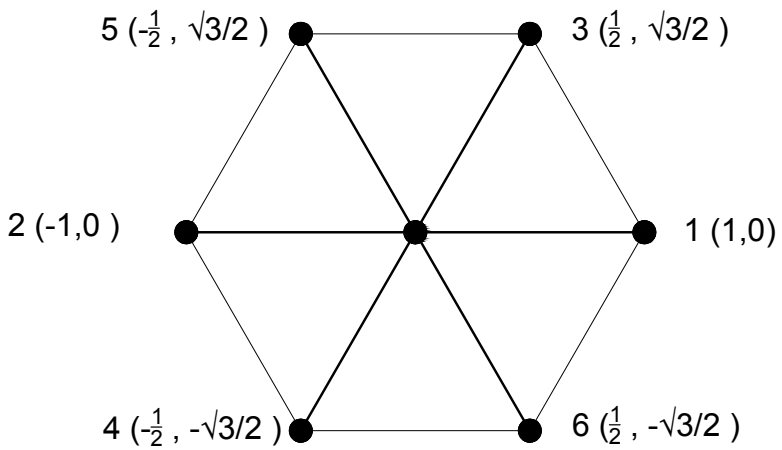


Figure 3.1: Bond numbering order and reference coordinates $\mathbf{X}$ of first nearest neighboring particles for the FCC lattice in 2D.

The linear elastic SPLM assumes a linear relationship between axial force associated with bond B_j and all of the bond stretches S_j associated with the bond list $\mathbb{B}$ (Gerstle, 2015).

Let us start by defining the state-based total stretch matrix $\{S^{total}\}$ in 2D and its relation with the classical strain matrix $\{\varepsilon\}$. The total stretch of a bond is divided into the elastic the stretch and the plastic stretch ($\{S^{total}\} = \{S^e\} + \{S^p\}$). When we use the stretch state or matrix, $\{S\}$, we are actually referring to the total stretch matrix.

The total stretch in a bond B_j is defined as:

$$S^{total}(B_j) = \frac{L^* - L_0}{L^*}, \tag{3.1}$$

where L^* (Equation 3.2) is the length of the bond between particle i and j in the deformed configuration, and L_0 (Equation 3.3) is the length of the bond in the reference configuration.

$$L^* \equiv \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{3.2}$$

$$L_0 \equiv \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2} \tag{3.3}$$

The total stretch state ($\{S_{total}\}$) of particle i is expressed as a matrix that contains the 6 bond stretches:

$$\{S^{total}\}_i \equiv \begin{pmatrix} S_1 \\ \vdots \\ S_j \\ \vdots \\ S_6 \end{pmatrix}_i . \tag{3.4}$$

If the stretch of all bonds is small enough, the bond direction for bond B_j is the vector of direction cosines, defined as $\{n_s\} = \left(\mathrm{Dir}\underline{\mathbf{X_R}}\right)\langle B_j\rangle$ or

$$\{n_s\}_i \equiv \begin{pmatrix} \frac{(X_j - X_i)}{L_0} \\ \frac{(Y_j - Y_i)}{L_0} \end{pmatrix}_i . \tag{3.5}$$

If the stretch is large, the bond direction unit vector is $\{n_L\} = \left(\mathrm{Dir}\underline{\mathbf{x_R}}\right)\langle B_j\rangle$ or

$$\{n_L\}_i \equiv \begin{pmatrix} \frac{(x_j - x_i)}{L^*} \\ \frac{(y_j - y_i)}{L^*} \end{pmatrix}_i . \tag{3.6}$$

The relation between the classical strain matrix and the SPLM stretch matrix is given by Equation 3.7. This relation only applies if the deformation field is spatially homogeneous.

$$\{S\} = [N]\{\varepsilon\} \tag{3.7}$$

or

$$\begin{Bmatrix} S_1 \\ \vdots \\ S_j \\ \vdots \\ S_6 \end{Bmatrix} = \begin{bmatrix} n_{X1}^2 & n_{Y1}^2 & n_{X1}n_{Y1} \\ & \vdots & \\ n_{Xj}^2 & n_{Yj}^2 & n_{Xj}n_{Yj} \\ & \vdots & \\ n_{X6}^2 & n_{Y6}^2 & n_{X6}n_{Y6} \end{bmatrix} \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} . \tag{3.8}$$

The $[N]$ matrix maps the classical strain matrix to the SPLM stretch matrix.
We can define the elastic stretch state as the total stretch state minus the plastic stretch state:

$$\{S^e\} = \{S^{Tot}\} - \{S^{Plast}\}. \tag{3.9}$$

The plastic stretch state will be defined in the next chapter.

The elastic SPLM constitutive relation is a function only of the deformation state: $\underline{\mathbf{T}} = \underline{\tilde{\mathbf{T}}}(\underline{\mathbf{x_R}})$. The force matrix, $\{T\}$, is similar to the stretch matrix:

$$\{T\}_i \equiv \frac{1}{2} \begin{pmatrix} F_1 \\ \vdots \\ F_j \\ \vdots \\ F_6 \end{pmatrix}_i . \tag{3.10}$$

The magnitude $F_i\langle B_j\rangle$ of the pairwise force $\mathbf{F_i}\langle B_j\rangle$ acting upon particle P_i due to interaction with its neighbor $\mathcal{N}[P_i]\langle B_j\rangle$ is obtained as:

$$F_i\langle B_j\rangle \equiv |\underline{\mathrm{T}}[P_i]\langle B_j\rangle| - |\underline{\mathrm{T}}[\mathcal{N}[P_i]\langle B_j\rangle]\langle B'_j\rangle|, \tag{3.11}$$

where B'_j is the opposite to B_j.

When a homogeneous stress field σ is applied to a lattice body, a relation between $\{\sigma\}$ and the SPLM force matrix $\{T\}$ can be derived. To find this relation the internal virtual work under kinematically equivalent deformations of both the classical model and the SPLM model must be equal:

$$\delta W_{classical} = \delta W_{SPLM}, \tag{3.12}$$

with

$$\delta W_{classical} = \lfloor\sigma\rfloor\{\delta\varepsilon\}\Delta V \quad \text{and} \tag{3.13}$$

$$\delta W_{SPLM} = \lfloor F\rfloor\frac{\lfloor L_i\rfloor}{2}\{\delta S\}. \tag{3.14}$$

Combining Equations 3.12, 3.13, 3.14 and 3.7 we get Equation 3.15:

$$\lfloor \sigma \rfloor \Delta V = \frac{1}{2} \lfloor F \rfloor [L_i][N]. \tag{3.15}$$

Solving for $\lfloor \sigma \rfloor$ leads to Equation 3.16:

$$\{\sigma\} = \frac{1}{2\Delta V}[N]^T[L_i]\{F\} = [M]\{F\}, \tag{3.16}$$

with $\Delta V = \frac{\sqrt{3}t_b L^2}{2}$, the volume of the 2D lattice particle, where t_b is the thickness of the lattice body, and

$$[M] \equiv \frac{1}{2\Delta V}[N]^T[L_i] \quad \text{or} \tag{3.17}$$

$$[M] = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} & \frac{\sqrt{2}}{8} \\ 0 & 0 & \frac{3\sqrt{2}}{8} & \frac{3\sqrt{2}}{8} & \frac{3\sqrt{2}}{8} & \frac{3\sqrt{2}}{8} \\ 0 & 0 & \frac{\sqrt{6}}{8} & \frac{\sqrt{6}}{8} & -\frac{\sqrt{6}}{8} & -\frac{\sqrt{6}}{8} \end{bmatrix} \tag{3.18}$$

For a spatially homogeneous small-strain field, the linear elastic constitutive relation is:

$$\{F\} = [K]\{S^e\}, \tag{3.19}$$

where $[K]$ is the micro-elastic SPLM stiffness matrix. From linear elasticity theory:

$$\{\sigma\} = [D]\{\varepsilon^e\}. \tag{3.20}$$

Multiplying Equation 3.16 by $[N]$ on both sides and combining it with Equations 3.7 and 3.16 we obtain Equation 3.21, which relates the classical constitutive matrix with the micro-elastic stiffness matrix.

$$[D] = [M][K][N] \tag{3.21}$$

$[K]$ is found by solving Equation 3.21 assuming that the two independent variables

are Young's modulus E and Poisson's ratio ν.

$$\begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{Bmatrix} = \begin{bmatrix} (a+b) & b & b & b & b & b \\ b & (a+b) & b & b & b & b \\ b & b & (a+b) & b & b & b \\ b & b & b & (a+b) & b & b \\ b & b & b & b & (a+b) & b \\ b & b & b & b & b & (a+b) \end{bmatrix} \begin{Bmatrix} S_1^e \\ S_2^e \\ S_3^e \\ S_4^e \\ S_5^e \\ S_6^e \end{Bmatrix} \tag{3.22}$$

Solving for a and b using Matlab yields:

$$a = \frac{2ELt_b}{\sqrt{3}(1+\nu)} \quad \text{and} \tag{3.23}$$

$$b = \frac{2ELt_b(1-3\nu)}{6\sqrt{3}(\nu^2-1)}. \tag{3.24}$$

3.2 Bond Peridynamic Lattice Damage Model

The elastic and plastic peridynamic models used in this thesis are state-based, which means that the force between two particles is a function of the state of all bonds inside the peridynamic material horizon δ of a particle. The Bond Peridynamic Lattice Damage Model (BPLDM) is based on the idea that in a particle, the "damage" process may be in different stages in each bond. This model is anisotropic because each bond can have a different value of damage.

Damage is modelled as a variable that represents the stiffness degradation in the material due to the breakage of the bonds at the meso level. The damage in the bond is introduced in the force state (Equation 3.19) using the following equation:

$$\{T_j^d\} = (1 - w_j)\{T_j\} \tag{3.25}$$

In the BPLDM the condition of damage initiation in a bond is that the bond force is greater than a critical bond force, which is a material property. The concept

of strain can not be applied in a lattice body. Instead, the concept of "Crack Opening Displacement", COD, is used. The COD is calculated in a bond B_j as:

$$COD(B_j) = S_e(B_j) \cdot L_j^*, \tag{3.26}$$

where $S^e(B_j)$ is the elastic stretch in bond B_j ($S^e(B_j) = S^{tot}(B_j) - S^{plast}(B_j)$ and $S^{tot}(B_j) = \frac{L_j^* - L_j}{L_j}$, with L_j^* the deformed bond length), and L_j is the undeformed length of bond B_j.

With COD computed, the damage parameter in a bond $w(B_j)$, is calculated using Equation 3.27. With this definition of the damage parameter, a particle can have a bond with no damage and another one completely damaged.

The damage parameter for a bond B_j is defined as:

$$w(B_j) = \begin{cases} 0.0 & \text{if} \quad 0 \leq COD \leq COD_0 \\ 1 - \frac{f_t'}{ES(COD_1 - COD_0)} \left[z(COD_1 - COD) + \gamma(COD - COD_0) \right] & \text{if} \quad COD_0 \leq COD \leq COD_1 \\ 1 - \frac{\gamma f_t'}{ES} \left[\frac{COD_c - COD}{COD_c - COD_1} \right] & \text{if} \quad COD_1 \leq COD \leq COD_c \\ 1.0 & \text{if} \quad COD \geq COD_c \end{cases} \tag{3.27}$$

where f_t' is the uniaxial tensile strength, z is a post-peak reduction factor, γ is the tension reduction parameter at the "knee" of the bilinear softening curve, E is the Young's Modulus, S is the total stretch in the bond ($S = S^e + S^p$), and COD_0, COD_1 and COD_c are the COD parameters.

Figure 3.2 shows the relation between the damage, w, and the COD. When the COD reaches the maximum tensile stretch in a bond ($\frac{\sigma_t}{E} L$), the damage w immediately jumps to a finite value, rather than starting from zero.

Before damage initiates the force state is computed with the state-based elastic equation ($\{F\} = [k]\{S\}$). When COD is greater than the maximum tensile stretch the force in each bond is calculated as:

$$F'(B_j) = (1 - w(B_j)) a S^e(B_j), \tag{3.28}$$

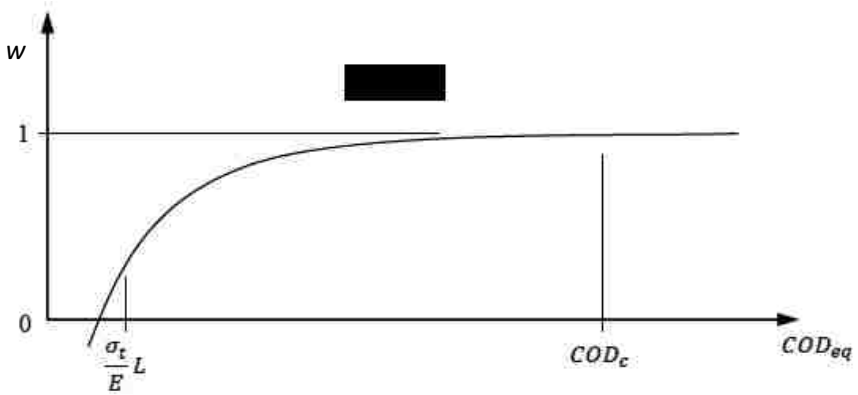


Figure 3.2: Damage w vs COD. From (Gerstle, 2015).

where a is the stiffness coefficient, computed in Equation 3.23. Note that the bond force is not a function of the two coefficients parameters a and b, as in the elastic region, but only of a. We made this assumption because b is more dependent on the Poisson's effect than a, and it is not clear how the Poisson's effect affects the response of the material when it is damaged.

We can now derive a equation for the effective bond force as function of the COD:

$$F'(B_j) = \begin{cases} aS_e + 6bS_{avg} & \text{if} \quad 0 \leq COD \leq COD_0 \\ \frac{2Lt_b}{\sqrt{3}(1-\nu)}\left[\frac{f'_t\left(z(COD_1-COD)+\gamma(COD-COD_0)\right)}{(COD_1-COD_0)} - ES\right] & \text{if} \quad COD_0 \leq COD \leq COD_1 \\ \frac{2Lt_b}{\sqrt{3}(1-\nu)}\left[\frac{\gamma f'_t(COD_c-COD)}{COD_c-COD_1} - ES\right] & \text{if} \quad COD_1 \leq COD \leq COD_c \\ 0 & \text{if} \quad COD \geq COD_c \end{cases} \tag{3.29}$$

where S^e is the elastic stretch of the bond, S_{avg} is the average stretch of the 6 particle bonds, t_b is the material thickness, and ν is the poisson's ratio. The first case represents the force in a undamaged bond. The second and third cases in Equation 3.29 represent the inelastic response of the material due the damage. Similar to the fictitious crack model (Hillerborg et al., 1976), outside the fracture zone, in the

BPLDM, when $COD \leq COD_0$ for any particle's bond, the constitutive model is linear elastic SPLM, and inside the fracture zone the COD equation governs the response of the material.

Figure 3.3 represents the stiffness degradation function of the material from

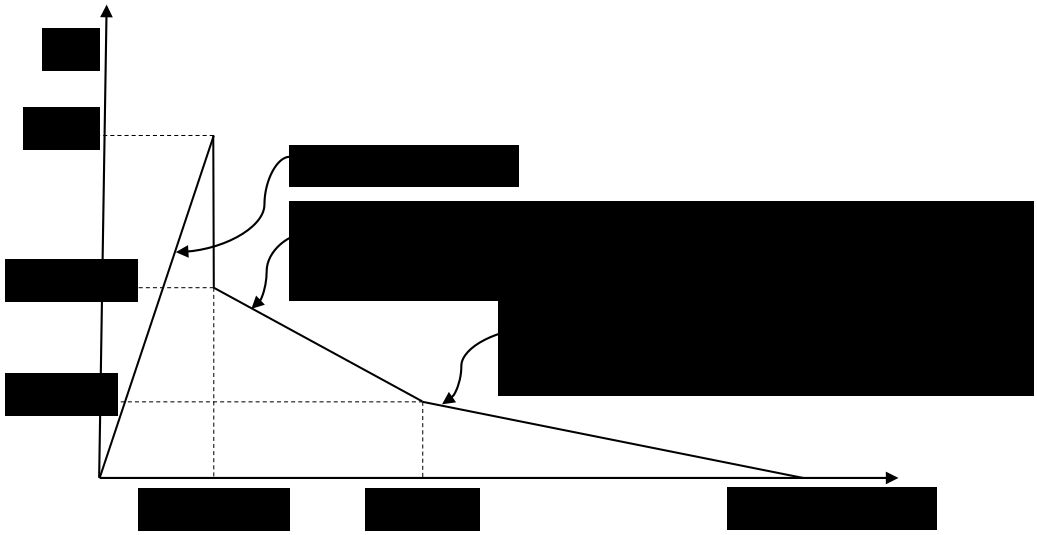


Figure 3.3: Stress vs. COD curve

Equation 3.29. As can be seen, subsequent to damage initiation, the greater the COD the smaller the effective bond force. The shape of the curve is similar to the function that Wittmann et al. (Wittmann et al., 1988) proposed, based on Hillerborg's fictitious crack model (Hillerborg et al., 1976). The parameters z and γ account for the stiffness reduction after cracking initiation. They can be set by the user, and in this model $z = 1.0$ and $\gamma = 0.25$. In addition, the parameters COD_0, COD_1 and COD_c are used to define the curve. These coefficients were calculated by Chapman (Chapman, 2011), so that the area under the curve approximates the energy absorbed, G_c, per new formed unit crack area (Hillerborg, 1976, Hillerborg et al., 1978; Wittmann et al., 1988; Bazant and Planas, 1998).

3.2.1 Tension Test in the BPLDM

The BPLDM is completely defined in the previous section. To test the model, it is necessary to check the dependency of the results on the material discretization, i.e., lattice spacing and rotation. One disadvantage of using finite element methods to model reinforced concrete is that some of its damage models are mesh sensitive, as we have seeing in Abaqus models. A convergence study is required, but the stresses and strains may not converge (ACI446, 2009). One of the objectives of this thesis is to present a model, using peridynamics, simpler than the FEA models and without their inconveniences. It is known that SPLM models may be sensitive to lattice rotation. To test the objectivity of the BPLDM the tension and compression tests were simulated for lattice rotations of 0°, 15° and 30°.

The direct tension test is sometimes employed to measure the uniaxial tensile strength of a specimen. For the simulations presented in this section, the geometry of the prismatic concrete specimen is 30 cm tall by 15 cm wide by 15 cm thick. The bottom of the specimen is fixed in all vertical degrees of freedom (boundary code 1), while the top layer of particles are the ones that apply the time-varying vertical displacement (boundary code 7). The maximum applied vertical displacement is set to 3 mm, in order to get a complete crack along the diameter of the cylinder.

SPLM is capable of simulating the loading under quasi-static or dynamic conditions. Quasi-static conditions, similar to the laboratory, were chosen by imposing a sufficiently long simulation time, and therefore, a smaller loading rate. Furthermore, the time step size is computed to be small enough to avoid instability. For all the simulations the number of time steps is computed as follows:

$$\text{End Time Step} = \frac{\beta \cdot \text{fundamental period}}{\text{critical time step}}, \tag{3.30}$$

where the *critical time step* is a function of the material properties that ensures the solution of the problem is stable, and the *fundamental period* is a function of the specimen's geometry. The factor β is chosen by the user, again, to ensure that the

loading rate is low enough to reproduce quasi-static loading conditions. A study on the loading rate effect in SPLM will be presented in the next chapter. For the tension and compression tests, it was found that a $\beta = 64$ is large enough to simulate quasi-static lab conditions. For other tests, like the Brazilian split cylinder, β is recommended to be around or higher than 128. However, as β increases, the simulation time grows too, hence, a balance must be found in order to keep the simulations feasible from an engineering point of view. As will be shown later in this thesis, the simulation time is a limitation for some problems.

Figure 3.4 compares the time history for the uniaxial tension test, with the same applied displacement at the top, for three lattice rotations (0°, 15° and 30°). It can be observed that the pre-peak response is linear in the three cases, obtaining the same slope, which means that the Young's modulus is not affected by the lattice rotation. The first difference that we find is the peak load between the maximum (0° - 99.176kN) and the minimum (30° - 71.507kN), which differs by a 27.9% (15° - 75.396kN). Hence, we can conclude that this model is sensitive to lattice rotation. In addition, the post-peak response for the 15° and 30° show that there is a hardening process, caused by some bonds in the transversal direction that are not fully damaged, due to their orientation.

In order to observe with more detail this hardening effect, Figure 3.5a shows the deformed shape before the second peak, and Figure 3.5c after that peak. Both figures show that there is a uniform distribution of damage, with low value, along the length of the specimen. It is important to point out the lateral displacement of the top block, which is the reason why there is a second load peak. If a closer view is taken, as in Figure 3.5e, it is seen that the bonds that were parallel to the applied load are unbroken. This phenomenon is due to the damage initiation and evolution approach, which is based on the uniaxial stretch of the bonds. In the undeformed configuration, due to the lattice rotation, the weakest direction is the transversal inclined 15°, since there are fewer bonds in that direction than in the others. The

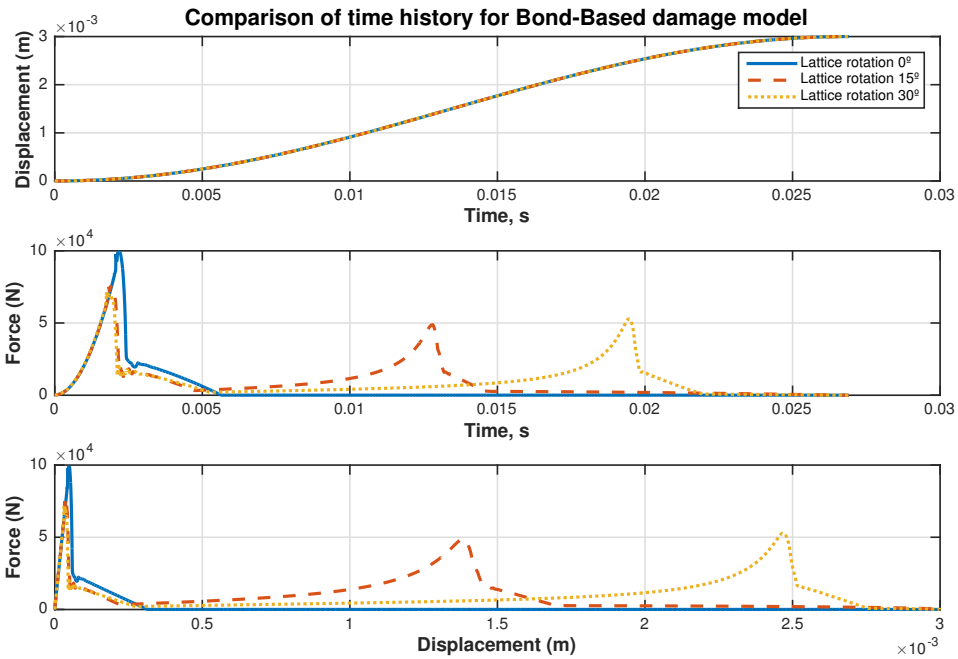
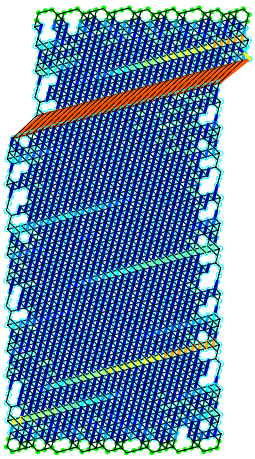


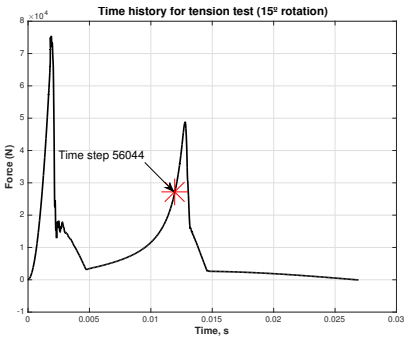
Figure 3.4: Time history plots for tension test bond-based damage model

same effect is observed when the lattice rotation is 30°, but the crack is now oriented at approximately 30° from horizontal (Figure 3.6c).

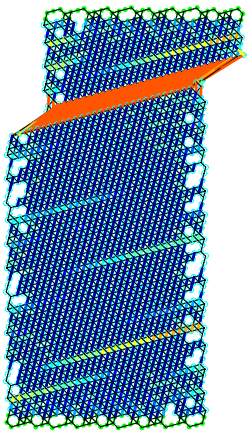
Finally, to conclude this study of lattice objectivity in the uniaxial tension test, Figure 3.6 shows the last time step for the three cases. When the lattice rotation is 15° and 30° the top block moves to right due to boundary imperfections, i.e., the boundary is not a perfect line of particles, since the hexagonal lattice is rotated, and some particles are out of the domain.



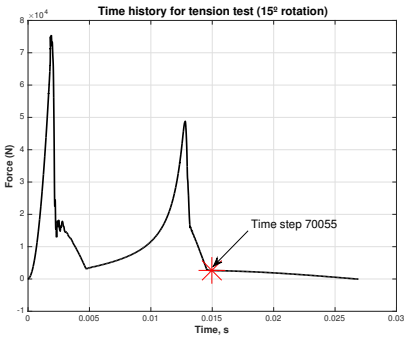
(a) Deformed shape at time step 56044



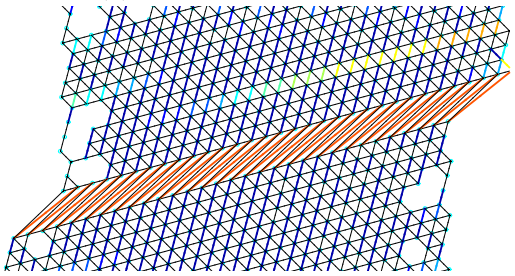
(b) Time history plot



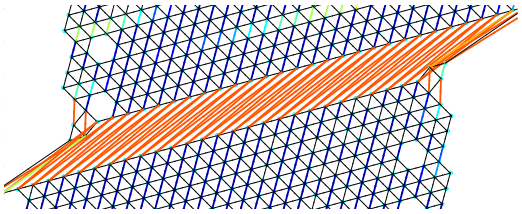
(c) Deformed shape at step 70055



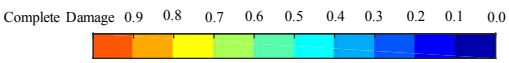
(d) Time history plot



(e) Deformed shape at time step 56044. Closer view of the crack.

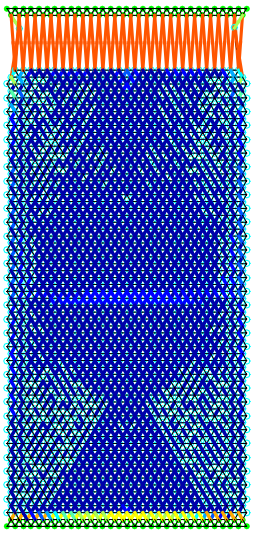


(f) Deformed shape at time step 70055. Closer view of the crack.

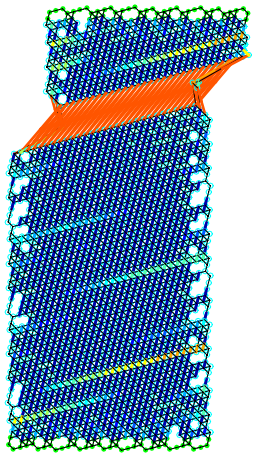


(g) Damage Scale Bar

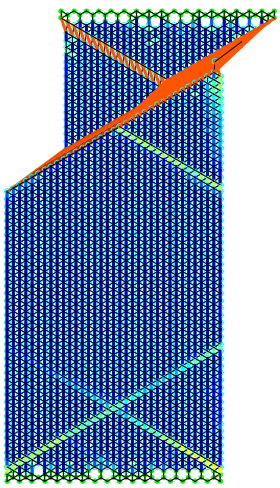
Figure 3.5: Deformed shape and time history of tension test with lattice rotation of 15°



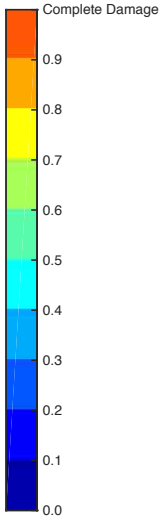
(a) Deformed shape at time step 126100 for lattice rotation of 0°



(b) Deformed shape at time step 126100 for lattice rotation of 15°



(c) Deformed shape at time step 126100 for lattice rotation of 30°



(d) Damage Scale Bar

Figure 3.6: Deformed shape comparison for tension test

3.2.2 Uniaxial Compression Test in the BPLDM

The uniaxial compression test is a very convenient laboratory test to measure the uniaxial compression strength of a specimen. The properties, geometry and simulation variables are the same as for the tension test, with the difference that the applied vertical displacement is now downwards instead of upwards.

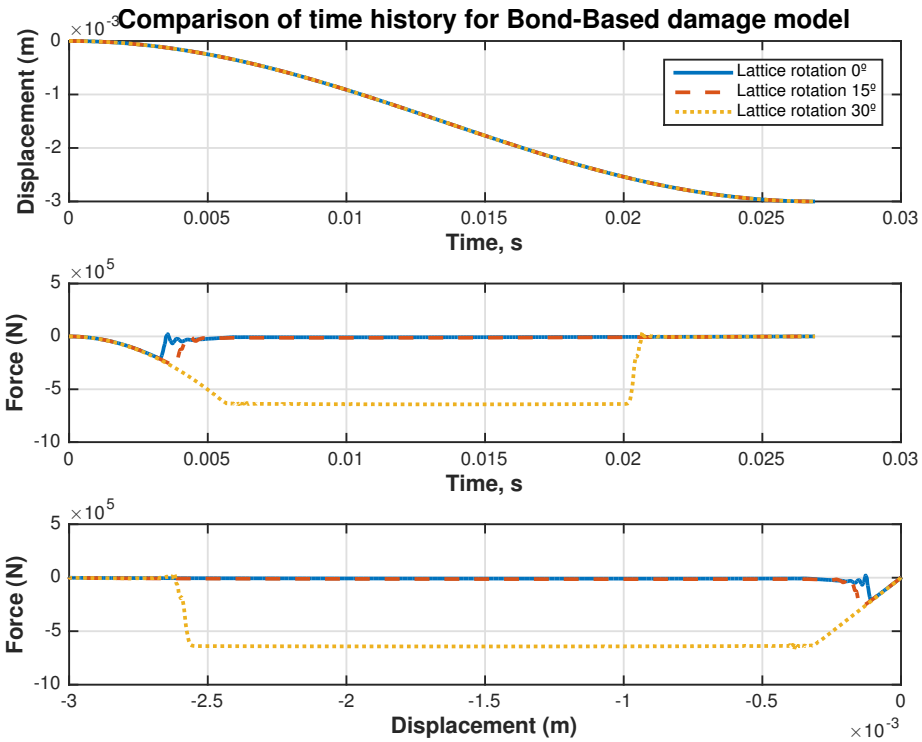
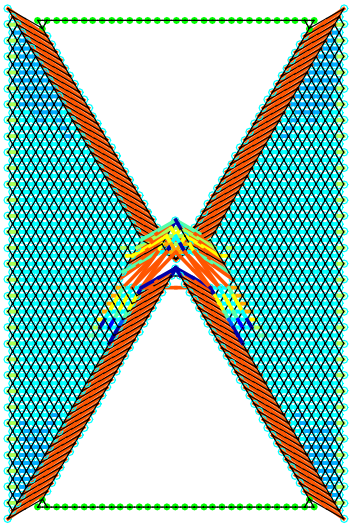


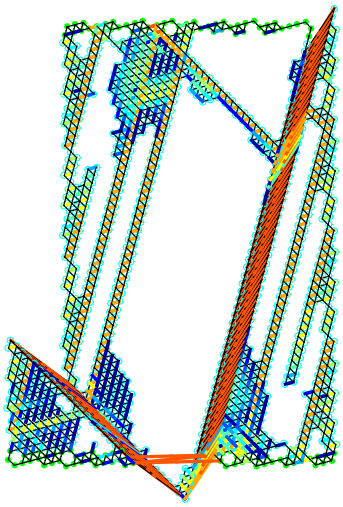
Figure 3.7: Time history plots for compression test bond-based damage model

Figure 3.7 presents the time history of the compression test for the three rotations (0°, 15° and 30°), and same applied vertical displacement. When the lattice rotation is 0° and 15° the force vs. displacement curve shows a similar maximum load, slightly higher for 15°. Nevertheless, when the lattice rotation is 30° the peak load is significantly higher. In addition, the plastic flow, characterised by the "plateau" in the force vs displacement curve, is governing the post-peak response of the specimen.

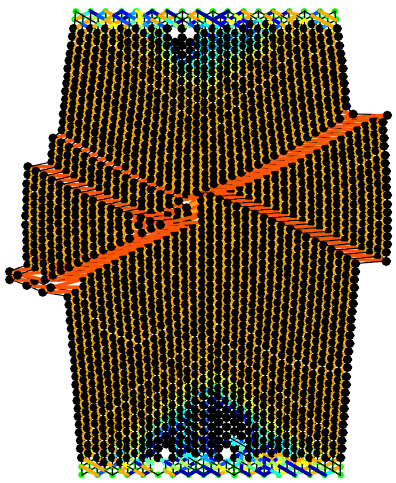
It is interesting to study the crack pattern when changing the lattice rotation. Figure 3.8 contains the deformed shape at the last time step. If we compare the three figures, we can see that the crack mechanism varies. For the first case (Figure 3.8a), there are two diagonal cracks, produced by the horizontal dilatation of the specimen. In Figure 3.8b, the weakest path is the slightly tilted crack, produced by the boundary imperfections. In the last case (Figure 3.8c), plasticity develops in the whole section before damage, then two wedges form, one at each side.



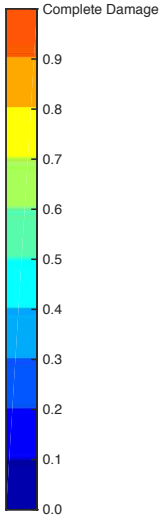
(a) Deformed shape at time step 126100 for lattice rotation of 0°



(b) Deformed shape at time step 126100 for lattice rotation of 15°



(c) Deformed shape at time step 126100 for lattice rotation of 30°



(d) Damage Scale Bar

Figure 3.8: Deformed shape comparison for compression test

3.3 New SPLM Damage Model

In the previous section, the BPLDM has been demonstrated not to be general enough to objectively model concrete structures. The dependency on the lattice rotation and boundary imperfections are the two main issues that need to be addressed. The State-Based Peridynamic Lattice Damage Model (SPLM damage) is developed here, whose objective is to eliminate the BPLDM problems. Thus, we move from a bond-based damage approach, where the damage in a bond is a function only of the bond stretch, to a state-based damage formulation, where the damage in a particle is a function of the state of its neighbouring particles.

The damaged force state is defined as: $\{T^d\} = (1-w)\{T\}$, or in matrix form:

$$\begin{Bmatrix} T_1^d \\ T_2^d \\ T_3^d \\ T_4^d \\ T_5^d \\ T_6^d \end{Bmatrix} = (1-w) \begin{bmatrix} a & 0 & 0 & 0 & 0 & 0 \\ 0 & a & 0 & 0 & 0 & 0 \\ 0 & 0 & a & 0 & 0 & 0 \\ 0 & 0 & 0 & a & 0 & 0 \\ 0 & 0 & 0 & 0 & a & 0 \\ 0 & 0 & 0 & 0 & 0 & a \end{bmatrix} \begin{Bmatrix} S_1^e \\ S_2^e \\ S_3^e \\ S_4^e \\ S_5^e \\ S_6^e \end{Bmatrix}. \tag{3.31}$$

The damaged force state is not a function of the stiffness coefficient b, as with the BPLDM.

In this model, similar to some continuum models (Simulia, 2015), the damage state is divided into two phases: 1) damage nucleation and 2) damage evolution. Damage nucleation, or initiation, occurs when the maximum principal stress at a particle, σ_1, is greater or equal to the uniaxial tensile strength of the material (F_t). This condition is similar to the Abaqus smeared crack and brittle crack models, with the difference that a second condition must be fulfilled. The second condition is the yield ratio, defined as:

$$\text{Yield Ratio} = \frac{J_2}{3F_t^2} \tag{3.32}$$

must be less or equal to 0.5. J_2 is the second invariant parameter (Equation 4.10) and F_t is the uniaxial tensile strength. This condition is included to separate tensile damage from plastic yield, which will be covered in next chapter. When these two conditions are fulfilled, a flag is turned on in the particle attributes list, and damage can start in the particle bond list. The expressions for σ_1 (Equation 4.6) and J_2 (Equation 4.10) in terms of the bond forces are developed in Section 4.1.

We introduced some modifications to the stiffness degradation function from the BPLDM. The fictitious crack model is applied again, with the same coefficients for the bilinear stress softening curve (Figure 3.3). Considering that we have chosen damage to be isotropic, instead of having a value of COD for each bond, we are introducing an equivalent crack opening displacement COD_{eq}. We can decide COD_{eq} to be a function of the maximum or minimum stretch in the particle. We consider that for the problems that we are modelling, it is more convenient to relate COD_{eq} with the uniaxial strain of a particle. This uniaxial strain ε_{axial}, is a function of the average elastic stretch S^e_{avg} (Gerstle, 2015):

$$\varepsilon_{axial} = S^e_{avg}\left(\frac{3}{1-2\nu}\right), \tag{3.33}$$

where ν is the Poisson's ratio. One can question the assumption that COD_{eq} does not consider the plastic component of the stretch state. For further research we should study the dependancy of tensile damage upon the plastic stretch. The implementation of a model that couples these two states may predict more accurately the response of concrete, but adds more complexity.

With ε_{axial} computed, COD_{eq} can be computed as:

$$COD_{eq} = \varepsilon_{axial} \cdot L. \tag{3.34}$$

We need to introduce some modifications to the damage and force equations, from the BPLDM derivation, to be consistent with the state-based formulation. By using COD_{eq} and S_{avg}, the damage is a function of all the particles bonds. We rewrite the

damage function as:

$$w(P_i) = \begin{cases} 0.0 & \text{if} \quad 0 \leq COD_{eq} \leq COD_0 \\ 1 - \frac{f'_t\left(z(COD_1 - COD_{eq}) + \gamma(COD_{eq} - COD_0)\right)}{ES_{avg}(COD_1 - COD_0)} & \text{if} \quad COD_0 \leq COD \leq COD_1 \\ 1 - \frac{\gamma f'_t}{ES_{avg}}\left[\frac{COD_c - COD_{eq}}{COD_c - COD_1}\right] & \text{if} \quad COD_1 \leq COD \leq COD_c \\ 1.0 & \text{if} \quad COD \geq COD_c \end{cases} \tag{3.35}$$

The bond force equation is:

$$F'(B_j) = \begin{cases} aS^e + 6bS_{avg} & \text{if} \quad 0 \leq COD_{eq} \leq COD_0 \\ \frac{2Lt_b}{\sqrt{3}(1-\nu)}\left[\frac{f'_t\left(z(COD_1 - COD_{eq}) + \gamma(COD_{eq} - COD_0)\right)}{(COD_1 - COD_0)} - ES\right] & \text{if} \quad COD_0 \leq COD_{eq} \leq COD_1 \\ \frac{2Lt_b}{\sqrt{3}(1-\nu)}\left[\frac{\gamma f'_t(COD_c - COD_{eq})}{COD_c - COD_1} - ES\right] & \text{if} \quad COD_1 \leq COD_{eq} \leq COD_c \\ 0 & \text{if} \quad COD_{eq} \geq COD_c \end{cases} \tag{3.36}$$

It is possible that a bond between two particles is completely damaged under tension, and then undergoes a compression force that repels both particles. According to Equation 3.36, when a bond is completely damaged it can not resist any force, therefore there is no force that prevents the two particles to overlap. To avoid this phenomenon, there is a condition in SPLM bond force calculation. If damage has initiated and the bond stretch is less than zero the bond force is computed as:

$$F'(B_j) = a \cdot S^e(B_j), \tag{3.37}$$

with a the SPLM stiffness coefficient, and $S^e(B_j)$ the bond elastic stretch. Figure 3.9 shows the diagram of force vs elastic stretch for tension and compression.

In the next two subsections, we demonstrate the new SPLM damage model using uniaxial tension and compression tests.

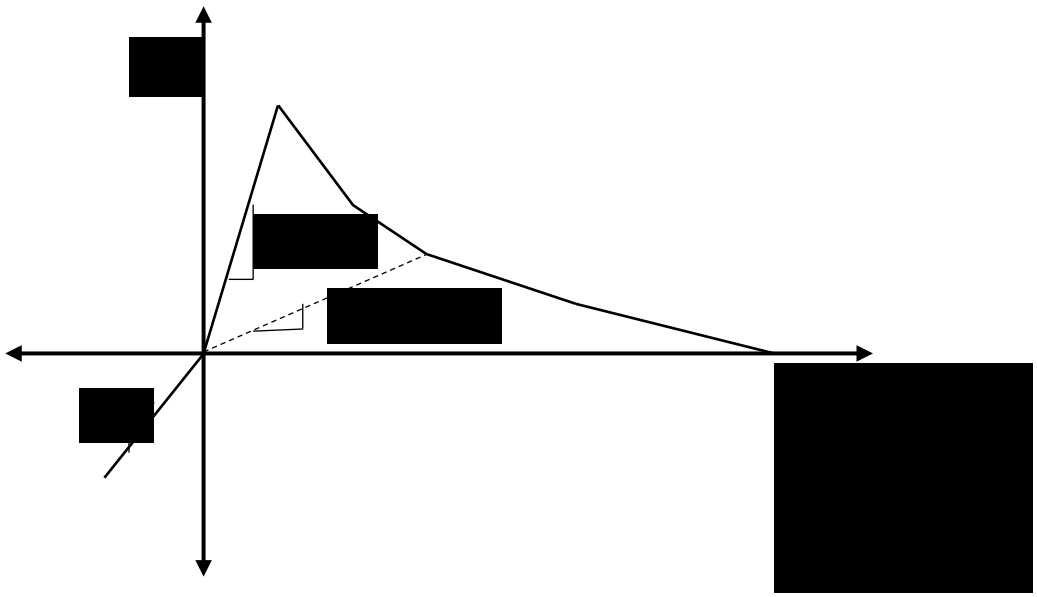


Figure 3.9: Force vs elastic stretch

3.3.1 Demonstration of the New SPLM Using the Uniaxial Tension Test

In this section lattice rotation is studied using the uniaxial tension test. The geometry and boundary conditions are the same as in the tension test presented in Section 3.2.1. The prismatic concrete specimen is 30 cm tall by 15 cm width by 15 cm thick.

In Figure 3.10, the time history for three lattice rotations (0°, 15° and 30°) is presented. The pre-peak response is perfectly linear and identical for the three rotations, while there are some small differences in the post-peak response due to the rotation. However, the peak loads are within a 8.62% difference ($58.120kN$ for 0°, $54.563kN$ for 15°, and $55.332kN$ for 30°).

If we take a look at the deformed shape at the last time step in Figure 3.11, it

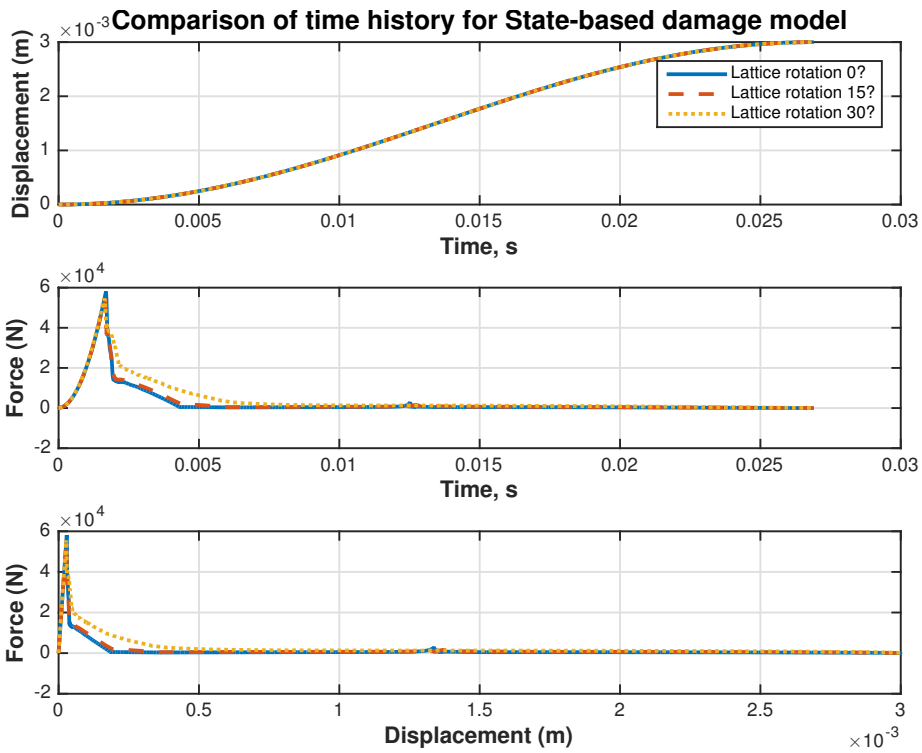
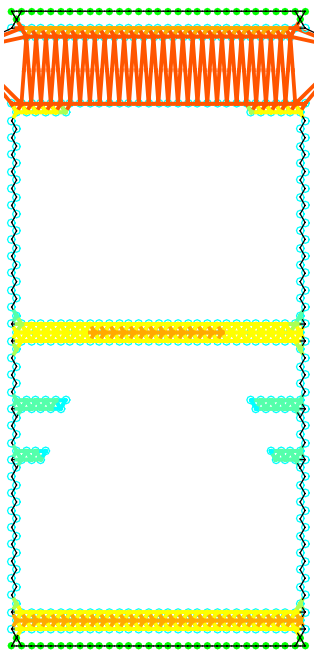


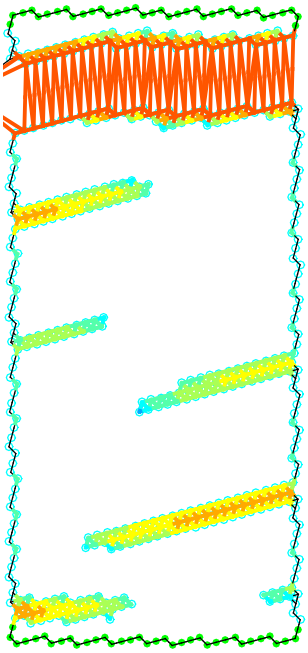
Figure 3.10: Time history plots for tension test using the new SPLM

shows that the crack pattern is more uniform than the BPLDM prediction. However, the crack direction is slightly dependent on the lattice rotation.

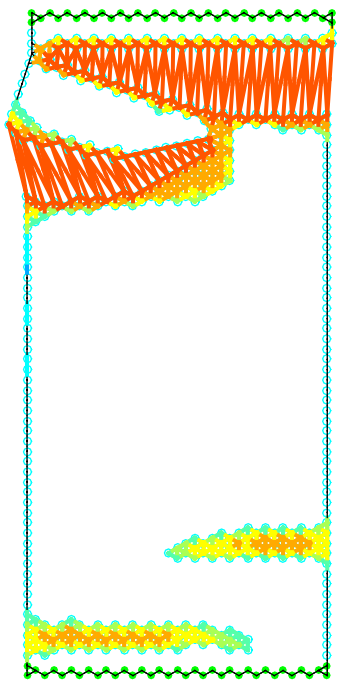
According to the classical theory, the peak load P is computed as $P = \sigma_t A$, with $\sigma_t = F_t = 2.758 \cdot 10^6 N/m^2$ and $A = 0.15 \cdot 0.15 = 0.0225 m^2$. Therefore, $P_{theoretical} = 62.068 kN$. The literature reports the tensile strength of direct tension specimens. Wright (Wright, 1955) presented the average strength for 32 concrete specimens. The average strength obtained was $\sigma_t = 275 psi = 1.89606 Mpa$. Thus, the peak load is computed as $P_{lab} = 1.89606 \cdot 10^6 N/m^2 \cdot 0.0225 m^2 = 48.349 kN$. With these results, we can conclude that the SPLM damage model provides a better estimation than the BPLDM. However, it is important to remark that the SPLM damage model prediction is slightly higher than the laboratory results.



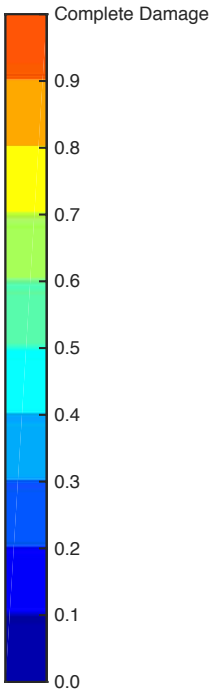
(a) Deformed shape at time step 126100 for lattice rotation of 0°



(b) Deformed shape at time step 126100 for lattice rotation of 15°



(c) Deformed shape at time step 126100 for lattice rotation of 30°



(d) Damage Scale Bar

Figure 3.11: Deformed shape comparison at last time step for tension test using the new SPLM

3.3.2 Demonstration of the New SPLM for Uniaxial Compression Test

To finish studying the new SPLM, a lattice rotation study using the uniaxial compression test is included. The geometry and boundary conditions are the same as in the compression test presented in Section 3.2.2. The prismatic concrete specimen is 30 cm tall by 15 cm width by 15 cm thick.

Figure 3.12 shows the comparison of the time history in the compression test. It can be observed that the peak load for the three rotations is the same. There is a difference regarding the plastic response, that is predominant when the lattice rotation is 30°. Furthermore, the softening process is different. We think this behaviour is due to boundary imperfections and a "particle interlock", similar to "aggregate interlock" in concrete.

Figure 3.13 shows the deformed shape for compression test specimens. For the three cases most of the particles have plastified (particles in black). The cracking pattern varies, slightly similar to BPLDM but more homogeneous. We can observe that there are still some dependencies on the boundary regarding the cracking direction. This phenomena is similar to the "wall effect" observed in concrete specimens.

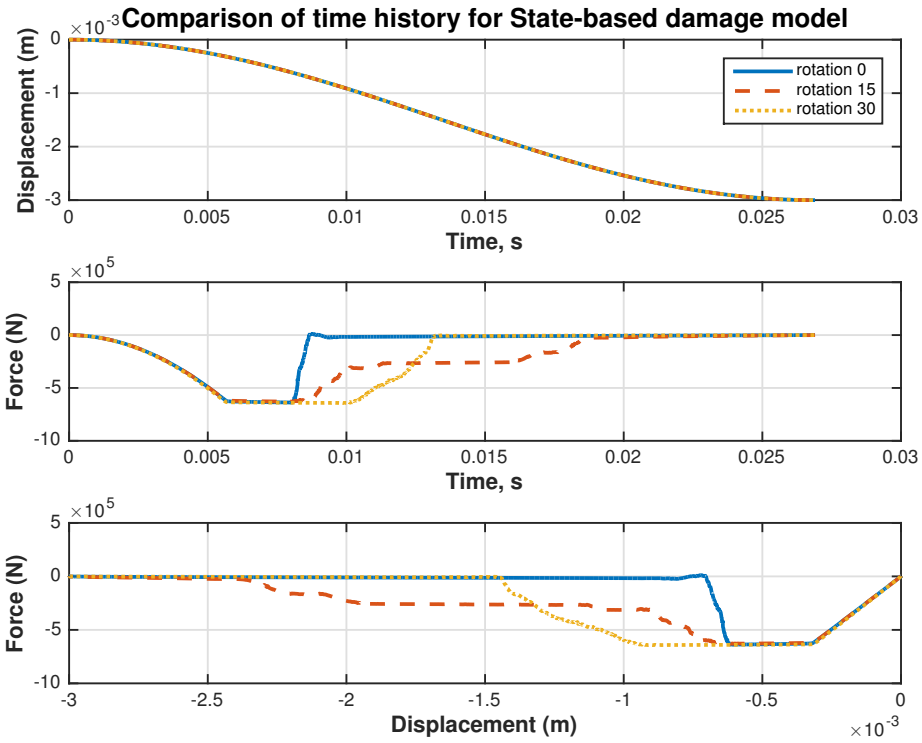
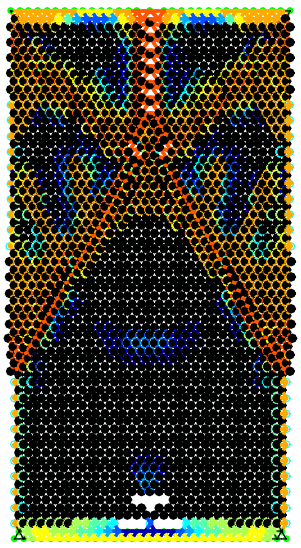
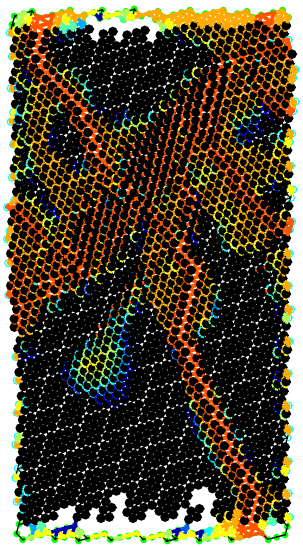


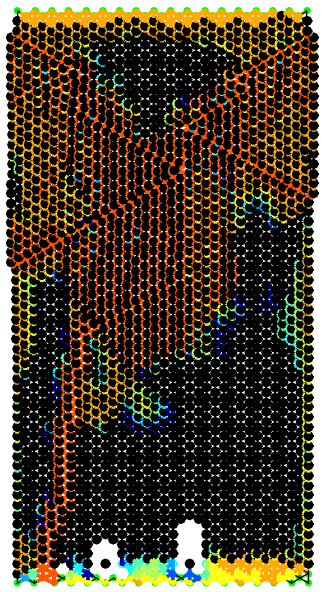
Figure 3.12: Comparison of time history for SBDM in the uniaxial compression test.



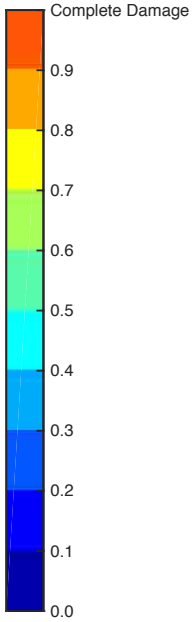
(a) Deformed shape at time step 126100 for lattice rotation of 0°



(b) Deformed shape at time step 126100 for lattice rotation of 15°



(c) Deformed shape at time step 126100 for lattice rotation of 30°



(d) Damage Scale Bar

Figure 3.13: Deformed shape comparison at last time step for compression test using the new SPLM

3.4 Conclusions

The BPLDM has been presented as a simple damage model. Due to its predictions of the uniaxial tension and compression tests, we conclude that the BPLDM model is nonobjective with respect to lattice rotation, and therefore not adequate to model concrete specimens. In addition, we have shown that, as a result of its bond-based approach, the model magnifies the boundary imperfections, predicting results that in some cases are not realistic. This model, applied to a lattice body that includes first and second nearest particles, may respond better to a homogeneous stress field. Nevertheless, adding more particles to the neighbour list increases the number of interactions between particles, and hence, needs more computations. This option should be explored in future research.

The issues regarding the BPLDM model justify the development of the new SPLM damage model. Maintaining the philosophy of the previous model, and this thesis, this new model presents a simple formulation that takes into account the neighbouring particles when computing the damage function. The SPLM damage model, has been shown to be relatively objective with respect to lattice rotation, but there are still some problems related to the cracking pattern and boundary imperfections that need to be corrected. The predictions of peak-load in the tension and compression tests show that the model reasonably replicates the laboratory results.

In conclusion, I think that the new SPLM damage model has been shown to be capable of modelling damage in concrete structures. We still need to explore the response of the model when applying more complex stress fields, as well as, the cases where the COD is not only a function of the elastic average stretch. But I conclude that, thanks to its simplicity and the reasonableness of the results, this model is quite reasonable for simulating concrete structures.

Chapter 4

SPLM Plasticity Damage Model

In this chapter, a new SPLM plasticity damage model is introduced. A rate dependency study is presented. The SPLM model is verified using the Brazilian split cylinder test. Finally, a comparison of the SPLM constitutive model with some continuum models is included.

4.1 New SPLM Plasticity Model

In this section we present a new SPLM plasticity damage model assuming plane stress conditions. The plasticity governing equations are derived. The SPLM plasticity model is a non-associated rate-dependent model. This model presents some modifications from Gerstle (Gerstle, 2015). The most important novelty (see [2]) is that the model accounts for damage due to excessive plastic flow.

4.1.1 Plastic Damage

Similar to classical theory of plasticity, the stretch matrix $\{S\}$ can be decomposed into the summation of the elastic stretch plus the plastic stretch. Thus, the elastic stretch is:

$$\underline{S^e}[P_i]\langle B_j\rangle = \underline{S}[P_i]\langle B_j\rangle - \frac{1}{2}\Big(\underline{S^p}[P_i]\langle B_j\rangle + \underline{S^p}[P_k]\langle B'_j\rangle\Big). \tag{4.1}$$

Gerstle (Gerstle, 2015) assumes that the plastic flow is caused by the deviatoric component of the force matrix, $\{F\}$, computed as:

$$\{F_{dev}\} \equiv \{F\} - \{F_{avg}\} \tag{4.2}$$

where $\{F_{avg}\}$ is equivalent to the hydrostatic stress ($\boldsymbol{\sigma_H}$) in classical theory. To derive the expression of J_2 in terms of the SPLM forces in plane stress conditions, we assume a 2D lattice where the six bond forces are reduced to three. Under a uniaxial stress acting upon the lattice body we can take the force in the direction of bonds 1 and 2 as the average of the bond forces 1 and 2, and so, with the other two pairs of bonds. Thus, we obtain the equivalent peridynamic stress tensor as:

$$\{\sigma_{equiv}\} = [M] \cdot \left\{ \begin{array}{c} F_1 \\ F_3 \\ F_5 \end{array} \right\}, \tag{4.3}$$

where $[M]$ (Equation 3.18) is the matrix that relates the classical stress matrix with the SPLM force state, and F_1 is the average force of bonds 1 and 2, F_3 is the average force in bonds 3 and 4, and F_5 is the average force in bonds 5 and 6. Solving Equation 4.3 we obtain the equivalent average normal stress in the plane in terms of the SPLM forces is:

$$\sigma_{AvgInPlane} = \frac{F_1 + F_3 + F_5}{\sqrt{3}Lt}, \tag{4.4}$$

where L is the lattice spacing and t is the thickness of the material. From Equation 4.3 we also calculate the maximum shear stress in the plane as:

$$\tau_{MaxInPlane} = \frac{\sqrt{F_1^2 + F_3^2 + F_5^2 - F_1F_3 - F_3F_5 - F_1F_5}}{\sqrt{3}Lt}. \tag{4.5}$$

With the stresses in the plane the principle stresses are computed using Mohr's circle:

$$\sigma_1 = max(\sigma_{AvgInPlane} + \tau_{MaxInPlane}, 0), \tag{4.6}$$

$$\sigma_2 = max(\sigma_{AvgInPlane} - \tau_{MaxInPlane}, 0), \tag{4.7}$$

$$\sigma_3 = min(\sigma_{AvgInPlane} - \tau_{MaxInPlane}, 0). \tag{4.8}$$

From $\{\sigma_{equiv}\}$, the average component of the force is computed, similar to $\boldsymbol{\sigma_H} = \frac{1}{3}(\sigma_1 + \sigma_2 + \sigma_3)\mathbf{I}$:

$$\{F_{avg}\} = \frac{\sqrt{3}(2F_1 + 2F_3 + 2F_5)}{9Lt}. \tag{4.9}$$

$\{F_{dev}\}$ is computed using Equation 4.2. In terms of the bond forces, the second invariant J_2 is:

$$J_2 = \frac{1}{2}\|\{F_{dev}\}\|^2 = \frac{4(F_1^2 + F_3^2 + F_5^2) - F_1F_3 - F_1F_5 - F_3F_5}{(Lt^2)/9} \tag{4.10}$$

The yield condition is reached when the norm of the deviatoric force surpass a critical value called $F_{devYield}$:

$$\|F_{dev}\| \geq F_{devYield}. \tag{4.11}$$

$\{F_{devYield}\}$ can be expressed in terms of the SPLM bond forces:

$$F_{devYield} = \frac{J_2}{3F_t^2} = \frac{4(F_1^2 + F_3^2 + F_5^2) - F_1F_3 - F_1F_5 - F_3F_5}{3(F_tLt)^2}, \tag{4.12}$$

where F_t is the uniaxial yield strength of the material.

When a particle is in yield condition, the plastic stretch can initiate and evolve.

The change in the plastic stretch matrix, $\left\{\Delta S_j^p\right\}$, at a given time step is computed as:

$$\left\{\Delta S_j^p\right\} = \Delta\lambda \frac{\{F_{dev}\}}{F_{devyield}}, \tag{4.13}$$

where $\Delta\lambda$ is a proportionality constant, similar to the constant in Levy-Mises flow equations ($d\varepsilon_{ij}^p = d\lambda\sigma_{Dij}$) that ensures isochoric plastic flow. Hence, the rate of plastic flow depends upon how far the force state lies outside the yield surface. Since the SPLM plasticity model is rate-dependent, it is not necessary to iterate within each time step to achieve equilibrium.

The proportional constant, $\Delta\lambda$ is modified from (Gerstle, 2015), since a flow parameter is introduced:

$$\Delta\lambda = \frac{\sigma_{yield}}{E}\left(\frac{\|F_{dev}\|}{F_{devYield}} - 1\right) \cdot MatFlowParam. \tag{4.14}$$

This $MatFlowParam$ parameter, set to 0.01 is a scaling factor, that in conjunction with the yield strain ($\frac{\sigma_{yield}}{E}$) has the objective of keeping the proportionality constant at the same order of magnitude as the yield strain, otherwise, $\Delta\lambda$ would be very small when the time step is very small.

This model includes a condition of damage due to plasticity. When the maximum plastic stretch in a bond, $max(\{S^p\langle B_j\rangle\})$, exceeds the maximum plastic stretch, *Material Plastic Damage Initiation*, the damage in the bond B_j is set to 1, which means that the bond is completely damaged.

Figure 4.1 shows the elastoplastic material response. When the deviatoric component of the force reaches the yield surface the material flows plastically. As we see in the figure, the response of the material is perfectly plastic. As defined previously, the yield surface does not evolve, and therefore does not account for hardening or softening phenomena.

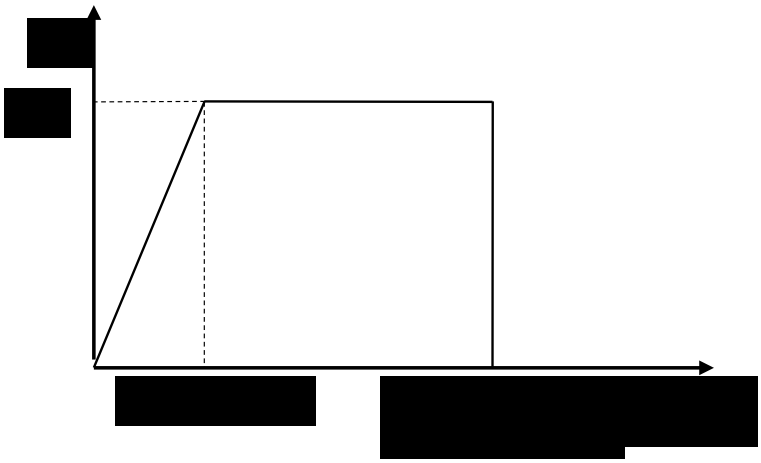


Figure 4.1: Elastoplastic response.

4.1.2 Rate Dependency

In this section, the Brazilian split cylinder is used to study the plastic flow dependency on the loading rate and the size of the time step. The Brazilian split cylinder is a commonly used laboratory test used to measure the tensile strength for plain concrete specimens. It is interesting to use this test in this section because a tensile stress is generated along the diameter of the cylinder, and concurrently, a compressive normal stress is generated in the section.

The specimen modelled with SPLM has a diameter of 15 cm and a thickness of 30 cm. Two steel loading plates of 2.5 cm width and 1.25 cm tall, transmit the applied displacement (0.5 mm) to the top and to the bottom of the cylinder.

Figure 4.2 shows the force vs displacement curves for the Brazilian split cylinder test. The applied displacement is fixed, while the simulation time varies from 32 fundamental periods to 256 periods. Thus, as the simulation time increases, the force is applied more slowly, and consequently, the loading rate decreases. The peak load decreases as the simulation time increases. This phenomenon, the fracture strain increment with an increment of the strain rate, has been reported in the literature

(ACI446.4R, 2004). The area under the force vs displacement curve grows with the simulation time, which means that the specimen fracture energy increases when loading is slower. The plastic flow grows as the simulation time increases, as shown in Figure 4.2. This response is due to the proportionality constant, which accounts for how far the deviatoric force is from the yield surface.

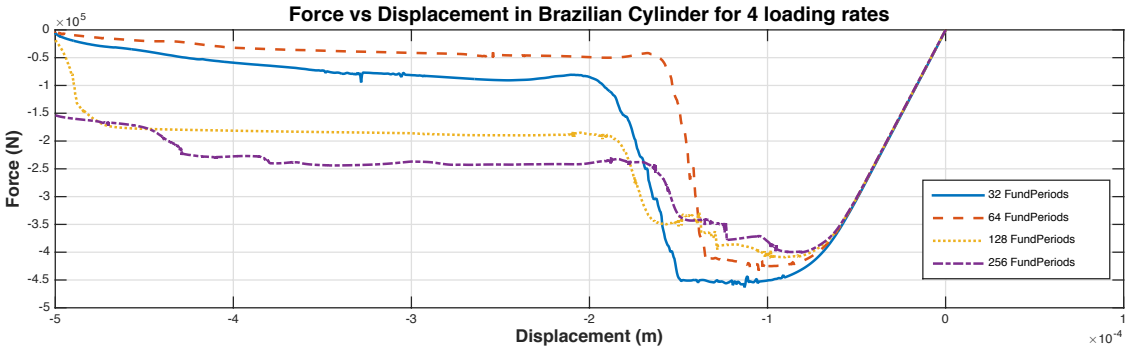
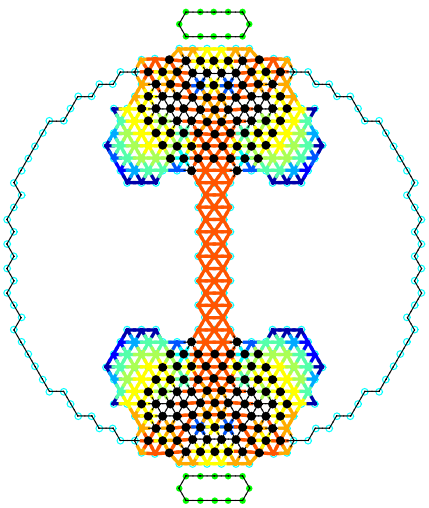


Figure 4.2: Force vs Displacement curves for the Brazilian split cylinder varying the simulation time

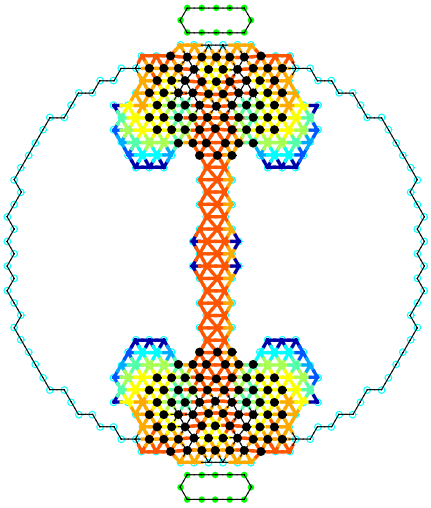
Figure 4.3 shows the deformed shape for the four simulation times. We can see that the number of particles that have plastified, increases slightly. In Figure 4.3d, we observe that the tensile crack has not propagated, instead, the particles near the loading plate have plastified.

Figure 4.4 shows the force vs time curves for the Brazilian split cylinder test when the size of the time step changes. The time step sizes are:

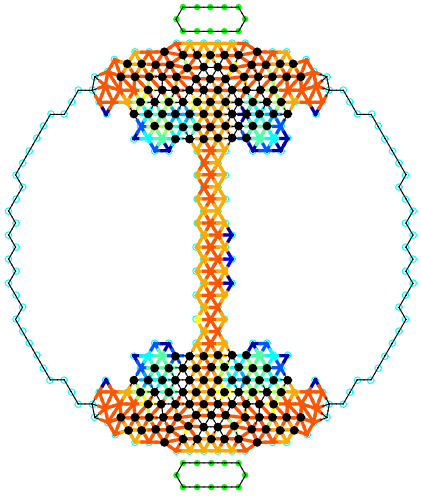
1. $6.74573777534234E-008s$.
2. $2.13318958683190E-007s$.
3. $3.01678564481074E-007s$.
4. $6.74573777534234E-007s$.



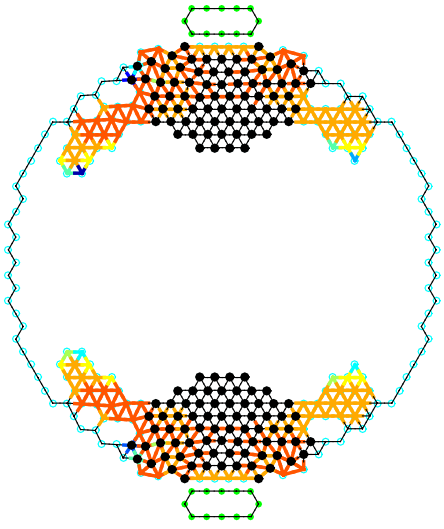
(a) Deformed shape for simulation time of 32 fundamental periods



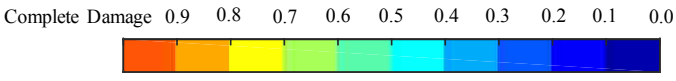
(b) Deformed shape for simulation time of 64 fundamental periods



(c) Deformed shape for simulation time of 128 fundamental periods



(d) Deformed shape for simulation time 256 fundamental periods



(e) Damage Scale Bar

Figure 4.3: Deformed shape of Brazilian split cylinder

We can observe that as the time step size increases, the plastic flow decreases, similar to Figure 4.2.

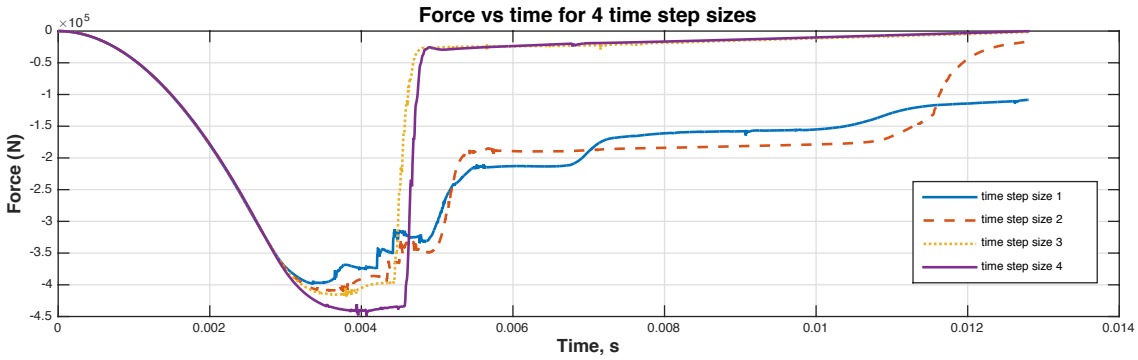


Figure 4.4: Force vs time for Brazilian split cylinder test when the size of the time step varies

Finally, to verify the results, we need to compare the peak load with the laboratory results. Assuming linear elastic behaviour, the split strength in the middle of the cylinder is

$$f_{sp} = \frac{2P_{max}}{\pi LD}. \tag{4.15}$$

Wright (Wright, 1955) reported that the average split strength for 32 specimens was $405psi = 2.792MPa$. Hence, the peak load is

$$P_{lab} = \frac{\pi LDf_{sp}}{2} = \frac{\pi \cdot 0.15m \cdot 0.3m \cdot 2.792MPa}{2} = 203.728kN. \tag{4.16}$$

The SPLM average prediction is approximately 400 kN, which is twice as the laboratory result. There are four possible sources of error:

1. The elastic SPLM.
2. The tensile damage SPLM.
3. The plastic damage SPLM.

4. The boundary conditions.

In order to completely verify the model we need to check that the boundary conditions and the loading rate match with the laboratory conditions. Furthermore, is difficult to calibrate the plastic and damage model since its parameters rely on the post-peak response of the specimen, which is not clearly reported in the literature due to the dynamic cracking nature and limitations of the data acquisition systems.

4.2 Comparison of SPLM with Continuum Models

With the elastic SPLM model and the new tensile SPLM damage model presented in the previous chapter, and the new SPLM plasticity model included in this chapter, we have the SPLM model for concrete completely defined. Now we can compare the SPLM model for concrete with the continuum models and Abaqus models presented in the literature review.

The objective of this thesis is to present SPLM as a viable alternative to continuum mechanics models, that without making non-realistic assumptions, and keeping the governing equations simple, can reasonably predict the response of a concrete structure.

As explained in Chapter 3, the tensile damage model is similar to the approach that Hillerborg et al. (Hillerborg et al., 1976) presented for the fictitious crack model, and similar to the smeared crack and brittle crack models in Abaqus. The Abaqus smeared crack model and plastic damage model have the fundamental problem of defining the stiffness degradation as function of the strain, which we have previously discussed. Moreover, these models define a parameter called "cracking strain", which, first does not make sense in consideration of the definition of strain, and second, it is

not objective. Our alternative is SPLM, that avoids this issue because the stiffness degradation function, or COD function, is a function of the bond stretches.

In Figure 4.5, a comparison of the force vs. time curves for the Brazilian split cylinder using SPLM and the Abaqus brittle cracking model is presented. The properties, geometry and loading conditions are identical. We can see that both models predict a slightly similar peak load. SPLM predicts a gradually decrease of the force after the peak-load is reached, due to the tensile cracking combined with the plastic flow. The Abaqus brittle crack model predicts a drop in the force after the peak-load, followed by a hardening that is not reasonable. However, we have to remember that this model does not include plasticity. This hardening effect could be caused by the elimination of elements.

Figure 4.6 shows the vertical displacements U_2 in the specimen. The mesh is composed of quadrilateral elements as shown in Figure 4.6a. The time step incrementation was set to 20, to save computation time. Figure 4.6b shows the first time step increment where the fully damaged elements are removed. As the crack grows, more elements are removed, as shown in Figure 4.6c. The tensile crack is predicted similarly to SPLM. This Abaqus model does not account for plasticity, hence that is why we do not see particles plastifying near the supports as expected, and shown in the SPLM simulations.

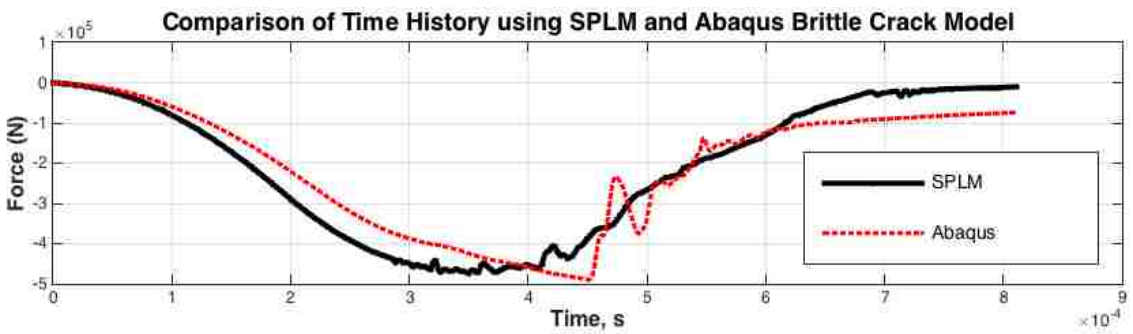
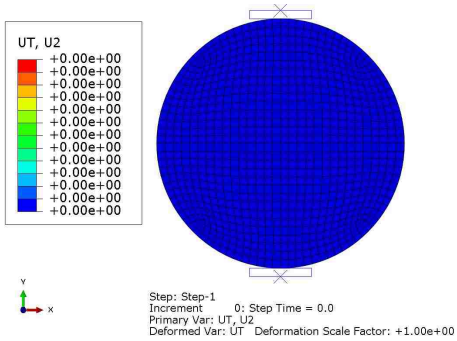
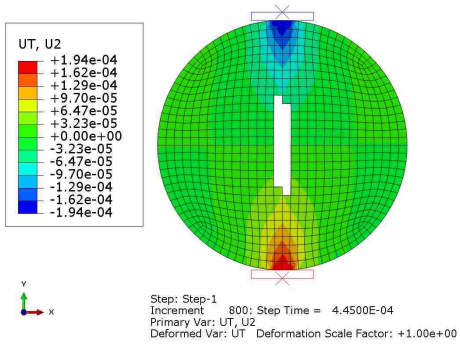


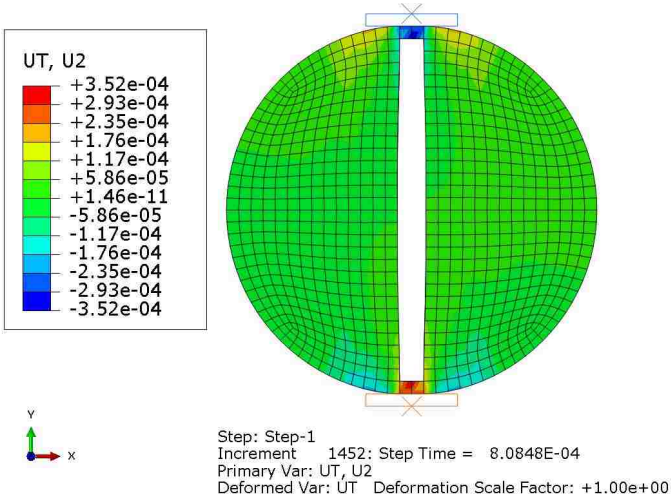
Figure 4.5: Comparison of force vs time for the Brazilian split cylinder using SPLM and Abaqus brittle crack model.



(a) Vertical displacement U_2 for time step 1.



(b) Vertical displacement U_2 after crack initiation.



(c) Vertical displacement U_2 for last time step.

Figure 4.6: Vertical displacements in the Brazilian split cylinder using Abaqus Brittle Crack Model.

The Abaqus plastic-damage model is based on Lubliner's and Lee and Fenves models (Lubliner et al., 1989; Lee and Fenves, 1998). These models present a constitutive relation that accounts for tensile damage and compressive crushing. Although the idea is plausible, the result of its implementation is inconvenient, since the model is too complex, and relies on too many parameters. Furthermore, their basic assumption of stress-strain constitutive behavior is not realistic. On the contrary, SPLM separates tensile damage and compressive crushing into two independent models.

One can argue that perhaps this assumption does not replicate the behavior of concrete when tensile damage and plasticity take place concurrently, and what therefore, there should be a dependency between these two SPLM models. However, we have found that despite this simplification, SPLM reasonably predicts the response of the Brazilian split cylinder, that has widely spatially varying stress states. It is difficult to calibrate the post peak response with the experimental data available. We can conclude that SPLM, with its own limitations and improvement potential, can compete with the Abaqus models, at least for the problems studied in this thesis.

The SPLM plasticity model has been shown to reasonably predict the response of a plain concrete specimen. I think that the damage due to plasticity function is perhaps oversimplified. But, contrary to Lubliner's models, where the internal variables (c and κ) are function of each other, SPLM can be easily modified, without reformulating the whole model, . SPLM uses the Mises yield criterion which is simple compared with the Drucker-Prager, Lubliner's or Lee's yield criteria. The yield surface in SPLM does not translate or evolve, due to damage, as in Lubliner's, Lee's and Abaqus plastic-damage models. This assumption in SPLM results in a much simpler equation for the yield area.

The following table contains a comparison of Abaqus and SPLM models.

	Ab. Smeared Crack	**Ab. Brittle Crack**	**Ab. Plast. Damage**	**SPLM**
Tensile damage	Hillerborg	Hillerborg	Lee and Fenves	Hillerborg
Compressive Damage	No	No	Lee and Fenves	Max. plastic stretch
No. of damage variables	1	1	2	2
Yield criteria	Mises Plasticity	N/A	Drucker Prager	Mises SPLM adapt.
Hardening	No	N/A	Yes	No

4.3 Conclusions

In conclusion, the new SPLM plasticity damage model has been shown to reasonably predict the response of plain concrete specimens. There are some simplifications and assumptions that may need to be changed, and there are some limitations to the model, such as the Brazilian split cylinder prediction. However, as a comparison with Abaqus models, for example, SPLM constitutive models have demonstrated that are capable of analysing concrete structures, without relying on a complex formulation.

Chapter 5

SPLM Reinforced Concrete Beams

The objective of this chapter is to present some modelling considerations of reinforced concrete beams using the new SPLM constitutive model. A new bond-slip model is also presented. Four three-point bending reinforced concrete beams are simulated with SPLM. The Canadian Code (CSA A23.3-04, 2004) and the ACI Code (ACI318.08, 2008) are chosen for comparing the strength prediction. The Canadian Code predicts the shear strength and captures the size effect better than the ACI Code. A size effect study is presented. Finally, the chapter concludes with a discussion of the accuracy of the results, the limitations of the model, and future research suggestions.

5.1 Bond-Slip Models

With SPLM, bond-slip modelling between concrete and rebar is essential to predict the response of reinforced concrete structures. Gerstle (Gerstle, 2015) defined the stiffness K of an individual SPLM bond between a concrete and a steel particle

as:

$$K = (1 - w_{concrete}) a_{concrete} \left(\frac{m_{steel}}{m_{concrete}} \right) S \tag{5.1}$$

where $a_{concrete}$ is the stiffness parameter for the concrete lattice body (Equation 3.23), $w_{concrete}$ is the damage of the concrete particle, m_{steel} and $m_{concrete}$ are the masses of the steel and concrete lattice particles, and S is the total stretch of the particle. The ratio $\frac{m_{steel}}{m_{concrete}}$ ensures that the elastic stiffness of the bonds connecting the steel body to the concrete body will not be so high that numerical instability arises. The peridynamic horizon (δ) between a concrete and a steel particle is assumed to be the maximum lattice spacing L of the steel and concrete lattices.

McVey (McVey, 2015) proposed a new expression for the bond stiffness, that does not account for the damage in the damage in the bond:

$$K = \alpha a_{concrete} \frac{min(m_{concrete}, m_{steel})}{m_{concrete}}, \tag{5.2}$$

where α is a parameter that defines the numerical stability of the bond.

We introduce a new modification in the calculation of K. Thus, the stiffness of bond connecting a steel particle with a concrete particle is denoted by

$$K[P_i]\langle B_j \rangle = \beta(1 - w) \cdot min(a(concrete), a(steel)), \tag{5.3}$$

where β is the bond-slip parameter, similar to α in McVey's model, set to 0.01, w is the damage parameter, taken as the maximum of the steel and concrete particles damage, and a is the stiffness parameter for the lattice body in plane stress conditions.

The force between a steel and a concrete particle in a bond $\langle B_j \rangle$ is computed as

$$F[P_i]\langle B_j \rangle = dc\langle B_j \rangle (K[P_i]\langle B_j \rangle S + linkForceDamp), \tag{5.4}$$

where $dc\langle B_j \rangle$ is the direction cosine vector of bond B_j, S is the total stretch between the two particles, and $linkForceDamp$ is the damping.

In this thesis, Equations 5.3 and 5.4 are used for the reinforced concrete simulations.

5.2 Beam Simulations

To test the applicability of the new SPLM constitutive models, a reinforced concrete beam was simulated in three-point bending and plane stress conditions. The test was repeated, changing only the steel ratio ($\rho_1 = 0.2\%$, $\rho_2 = 0.5\%$, $\rho_3 = 1\%$ and $\rho_4 = 2\%$). The geometry of the beam is presented in Figure 5.1. The longitudinal reinforcement was originally modelled as a single equivalent bar, that concentrates the total steel area. For these simulations, we decided to divide that area in two layers, thus, the number of bonds between concrete and steel particles increases and bond-slip effects can be minimized.

The vertical displacement $\Delta(t)$ applied at midspan is assumed as:

$$\Delta(t) = \frac{\Delta_{max}}{2}\left(1 - \cos\frac{\pi t}{t_{EndRamp}}\right), \tag{5.5}$$

where $\Delta_{max} = 0.01m$ is the maximum vertical applied displacement, and $t_{EndRamp}$ is the simulation time when the applied force is constant. The end ramp time is set by the user to the 90% of the simulation end time, which means that in the last 10% of the simulation, the displacement is constant.

The material properties are presented in Table 5.1, for concrete, and in Table 5.2, for the steel rebars and loading plates. Table 5.3 contains the parameters used in SPLM. Each test had 8225 particles, a simulation time of 0.16 s, 459400 time steps and took approximately 4 hours running on four processors.

We use the Canadian Code (CSA.3-04, 2004) and the ACI code (ACI318-08, 2008) as a comparison with the SPLM predictions. The flexural beam strength prediction using the CSA and ACI is computed in Equation 5.6 for this first case

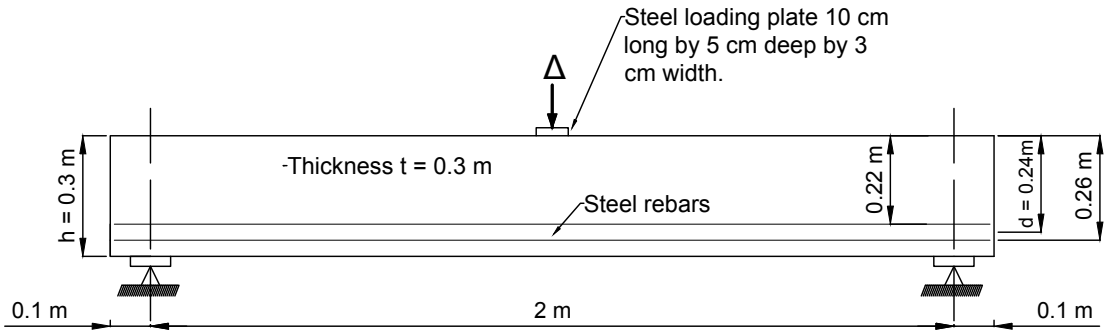


Figure 5.1: Geometry of reinforced concrete beam analyzed.

Parameter	SI Value
Compressive strength, F'_c	27.58 MPa
Young's Modulus, E	24.86 GPa
Poisson's ratio, ν	0.20
Mass density, ρ	2323 kg/m^3
Lattice spacing, L	2 cm
Internal damping ratio, $\xi_{internal}$	0.20
Tensile strength, F_t	2.758 MPa
COD at "knee" of bilinear curve, w_1	0.026 mm
Ultimate tensile COD, w_c	0.26 mm
Tensile damage parameter, γ	0.25

Table 5.1: Material properties for plain concrete.

Parameter	SI Value
Yield strength, F_y	413.7 MPa
Young's modulus, E	200 GPa
Poisson's ratio, ν	0.3
Mass density, ρ	7850 kg/m^3
Lattice spacing, L	1.0 cm
Damping ratio, ξ	0.2

Table 5.2: Material properties for steel rebars.

SPLM Parameter	Value
Plastic flow parameter	0.01
Material stretch at which plasticity causes total damage	0.002
Bond slip parameter	0.01

Table 5.3: SPLM material properties.

where $\rho = 0.2\%$:

$$
\begin{aligned}
&A_s = \rho bd = 0.002 \cdot 0.3m \cdot 0.24m = 1.44 \cdot 10^{-4} m^2, \\
&a = \frac{\phi_s A_s f_y}{\phi_c \alpha_f f'_c b} = \frac{1 \cdot 1.44 \cdot 10^{-4} m^2 \cdot 413.7 MPa}{1 \cdot 0.85 \cdot 27.58 MPa \cdot 0.3m} = 7.1 \cdot 10^{-4} m, \\
&M_n = A_s f_y (d - a/2) = 1.44 \cdot 10^{-4} m^2 \cdot 413.7 \cdot 10^6 N/m^2 (0.24m - 7.1 \cdot 10^{-4} m/2) = \\
&\quad 11.739 kN \cdot m, \\
&P_{nbending} = \frac{4M_n}{L} = 23.4787 kN,
\end{aligned}
\tag{5.6}
$$

where ϕ_s and ϕ_c are the material safety factor, assumed to be 1 because we are "comparing" the design code prediction with a simulation prediction, and α_f is the angle between the shear friction reinforcement and shear plane, which is equal to 0.85 according to the code (CSA.3-04, 2004).

The shear strength prediction using the CSA's simplified method is:

$$
\begin{aligned}
&\beta = \frac{230}{1000 + d_v} = \frac{230}{1000 + 0.9 \cdot 240} = 0.1855, \\
&V_n = \phi_c \lambda \beta \sqrt{f'_c} b d_v = 1 \cdot 1 \cdot 0.1855 \sqrt{27.58} \cdot 300mm \cdot 240mm = 70.1352 kN, \\
&P_{nShearCSA} = 2V_n = 143.039 kN.
\end{aligned}
\tag{5.7}
$$

ACI predicts a shear strength of:

$$
\begin{aligned}
&V_n = 2\lambda \sqrt{f'_c} b d = 2 \cdot 1 \sqrt{4000 psi} \cdot 6894.8 Pa/psi 0.3 \cdot 0.24 = 62.78 kN, \\
&P_{nShearACI} = 2V_n = 125.58 kN.
\end{aligned}
\tag{5.8}
$$

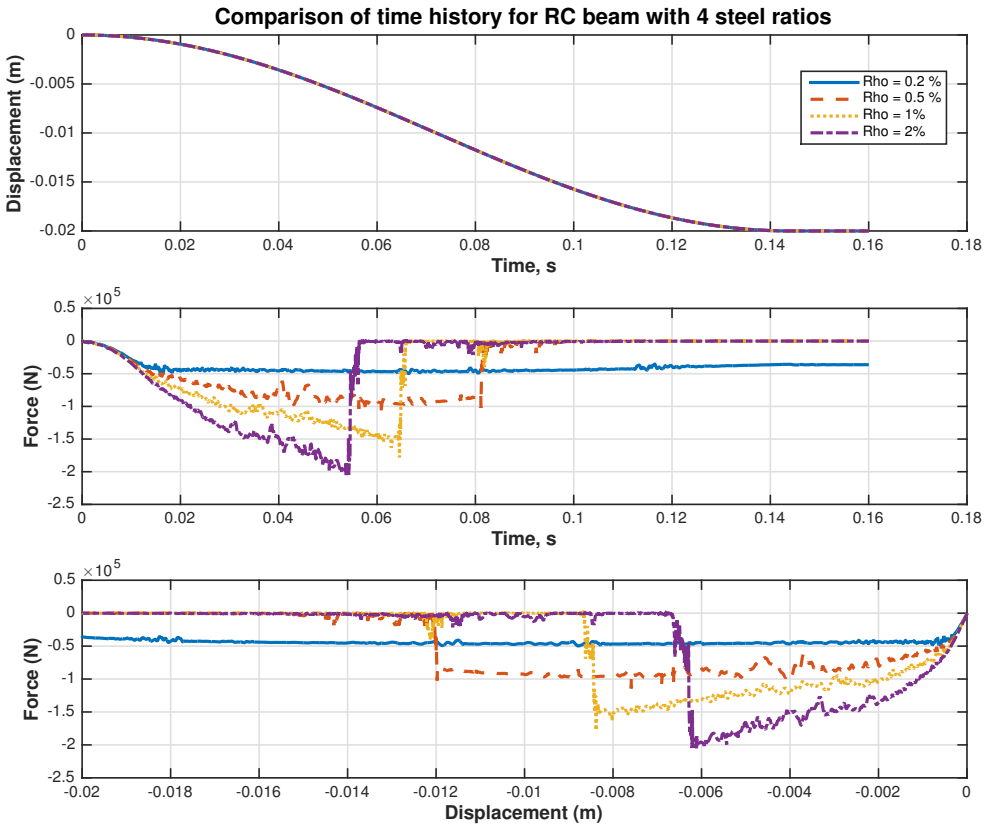


Figure 5.2: Comparison of time history for a reinforced concrete beam with 4 steel ratios.

Figure 5.2 shows the time history for the three-point bending beam modelled with four different steel ratios. As we observe, the capacity of the specimen, measured as the maximum applied load, increases as the reinforcement increases.

The unmagnified deformed shape at the last time step, for $\rho_1 = 0.02\%$ case, is shown in Figure 5.3. The specimen is failing in bending, due to yield of the reinforcement. For all the figures in this chapter, the colours code is: grey for steel particles that have yielded, black for concrete particles that have plastified, and from blue (low damage value) in a scale to red (completely damaged), the damaged

concrete particles. There is a "clean" vertical crack at midspan, under the loading plate, and two additional cracks are propagating. The steel rebars yield at the same time small cracks, in the tension side, are growing. SPLM predicts a maximum load of 49.810 kN, while the CSA and ACI predict a maximum load of $P_{nBending} = 23.478$ kN in bending. Figure 5.4 shows the unmagnified deformed shape for the beam

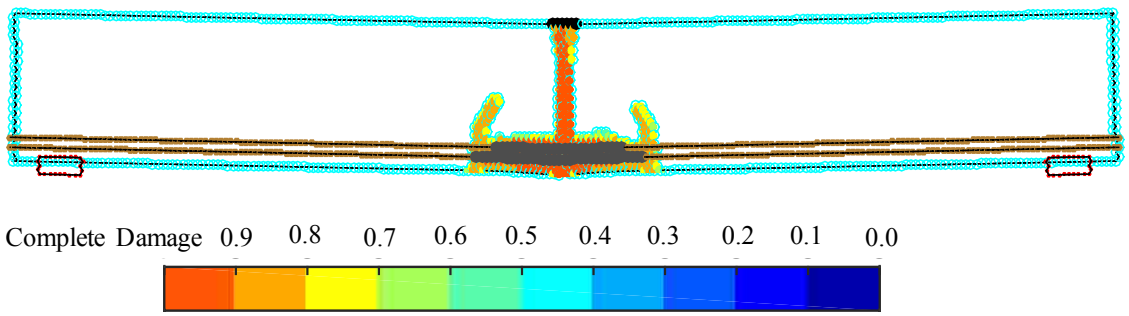


Figure 5.3: Unmagnified deformed shape RC beam with $\rho = 0.2\%$.

with $\rho_2 = 0.5\%$. There is non uniform plastic deformation along the rebars. This effect is perceptible in the displacement vs force curve (Figure 5.2), where one can see that when the specimen reaches it maximum elastic strength, the force keeps approximately constant when increasing the displacement until it completely fails. Two shear cracks propagated completely, producing an important debonding at the left side. Furthermore, there is compressive crushing at the top, under the loading plate, I think due to the loading rate. For this case, SPLM predicts a maximum force of 114.244 kN, while the Canadian Code predicts a maximum load in bending of $P_{nBending} = 57.382$ kN. If the steel ratio is increased to $\rho_3 = 1\%$ we clearly see that the concrete is governing the maximum capacity of the beam. Figure 5.5 shows that the failure mechanism is shear cracking and debonding of the rebar. In this case, the rebars barely plastify, but they are debonded in both sides of the beam. Also notice that the shear cracks that propagate near the rebars, go to the top of the beam near the supports. The Canadian Code predicts a maximum bending load

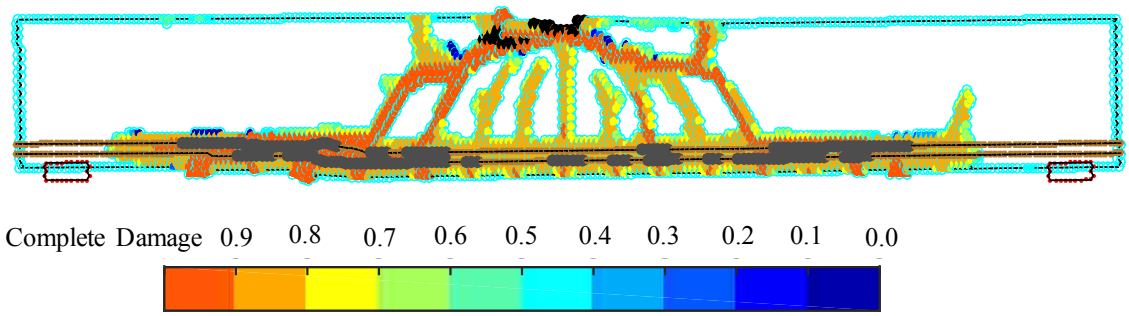


Figure 5.4: Unmagnified deformed shape RC beam with $\rho = 0.5\%$

of $P_{nBending} = 110.384$ kN and shear capacity of $P_{nShearCSA} = 143.038$ kN which is interesting because, although the equations assume the bars to be fully developed, the beam is failing due to shear cracking. The ACI code predicts a shear strength of $P_{nShearACI} = 125.58kN$. Finally, the SPLM prediction is 177.910 kN, which again, is higher than the code prediction. To finish with this section we are discussing

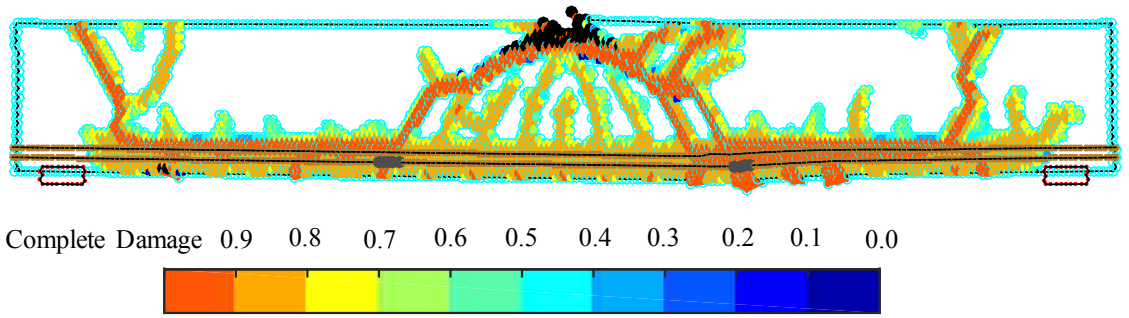


Figure 5.5: Unmagnified deformed shape RC beam with $\rho = 1\%$

the last case, where $\rho = 2\%$. Figure 5.6 shows a cracking pattern similar to the previous case. The compressive crushing under the loading plate is more significant in this case. The time history plot (Figure 5.2) shows quite significant "vibrations", which are due to all the cracks that are initiating and propagating. SPLM predicts a maximum force of 205.084 kN, while the Canadian Code predicts a pretty close bending load of $P_{nBending} = 203.248$ kN and shear capacity of $P_{nShearCSA} = 143.0388$

kN, and the ACI $P_{nShearACI} = 125.58kN$.

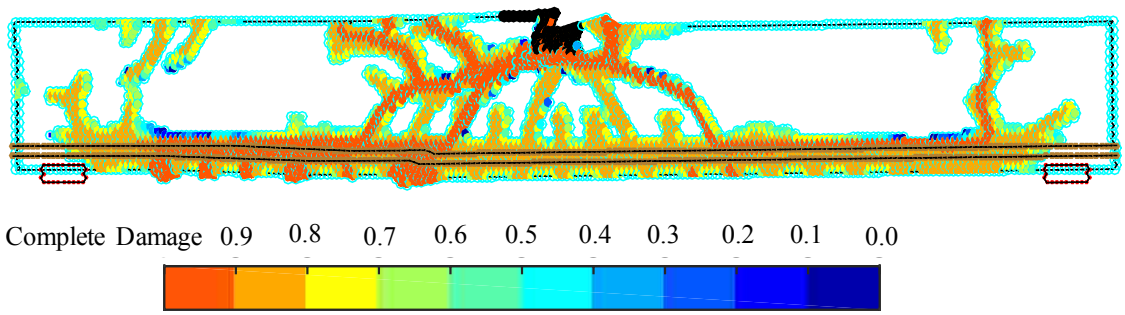


Figure 5.6: Unmagnified deformed shape RC beam with $\rho = 2\%$

5.3 Size Effect Study

Size effect is defined by Bazant (Bazant and Planas, 1998) as the reduction of the structural strength, produced by a change of structure size, of the actual load capacity of a structure from the load capacity predicted by plastic limit analysis. In the design of a big structure, since there is more energy stored, fracture mechanics analysis is essential, besides stress analysis. In this section we repeat the three-point bending beam test with $\rho = 1\%$, now incrementing the beam size.

Due to time limitation only three simulations were performed. The span, height and applied load are multiplied by a scalar factor, but the thickness remains constant ($b = 0.3m$). The computational capacity has been a limiting factor, therefore, only three beams with a scale factor of 1, 4 and 6 were simulated. For the beams with a scale factor of 4 and 6 the simulation were stopped after 20 hours running in the supercomputer. The beam with scale factor of 4, was run on 16 processors, and the one with scale factor of 6 was run on 32 processors. It was not possible to output a deformed shape figure since the beams had 112,000 and 305,000 particles respectively.

Figure 5.7 shows the force vs displacement curve for three beams. Comparing

the peak load of the smallest beam with the largest, one can say that indeed, the SPLM is predicting a size effect, since the load is not six times higher. However, I can not conclude that this statement is completely right. Figure 5.8 shows the size effect plot, where the y-axis is the nominal strength calculated as $\sigma_n = \frac{P_n}{bD}$, where b is the thickness and D is the span. Note that b is constant and equal to 0.3 m for the three cases. The results show that there is a size effect prediction, since the nominal strength decreases with the structure size.

When at the same time we increase the size of the problem, the simulation time is increased by the same factor. However, Figure 5.7 shows that the vibrations due to cracking are much significant when the size factor is 6. Therefore, I can not conclude that the reduction of predicted strength with respect to the stress analysis solution, is purely due to a size effect or the dynamic cracking response governs the problem.

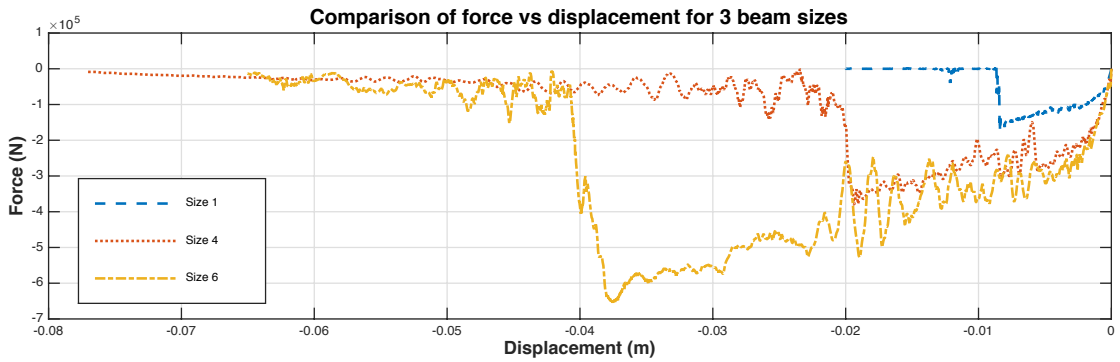


Figure 5.7: Comparison of force vs displacement for a RC of 3 sizes with $\rho = 1\%$.

When the scale factor is 4, the ACI predicts a shear strength of:

$$\begin{aligned} V_n &= 2\sqrt{4000} \cdot 6894.8 Pa/psi \cdot 0.3m \cdot 1.14m = 298.27kN \\ P_{nShearACI} &= 2 \cdot 298.27kN = 596.54kN. \end{aligned} \tag{5.9}$$

The CSA predicts a maximum bending strength of $P_{nBending} = 941.9511$ kN and a shear strength of $P_{nShearSimplified} = 373.251$ kN with the simplified method.

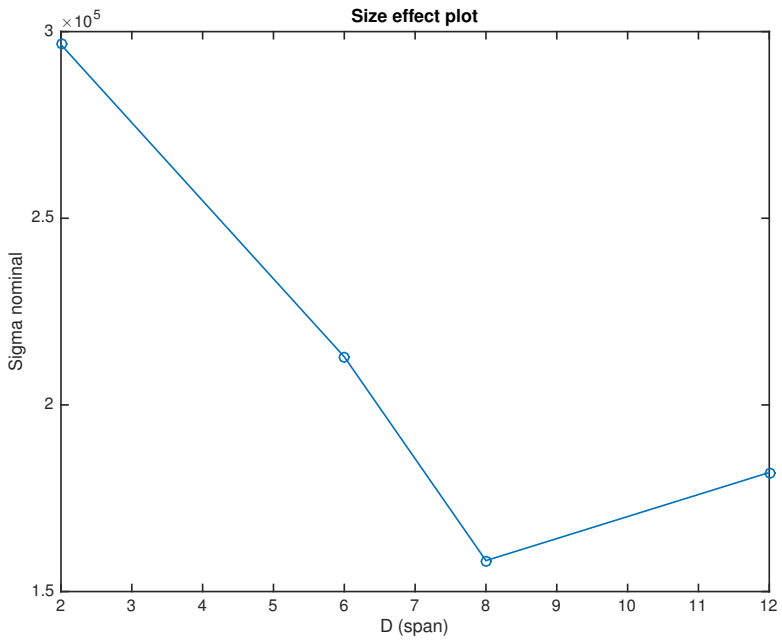


Figure 5.8: Size effect plot

The shear strength prediction using the general method, that captures the size effect, is computed next.

First we calculate the moments and shear loads:

$$M = M_{sw} + M_{load} = \frac{(\gamma_c bh)L^2}{8} + PL/4;$$
$$V = V_{sw} + V_{load} = (\gamma_c bh)L/2 + PL/2.$$

The size factor and strain are

$$s_{ze} = \frac{35 s_z}{15 + a_g};$$
$$\varepsilon = \frac{(M)/d_v + V}{2E_s A_s}. \tag{5.10}$$

We then express β in terms of the maximum load P, and substitute s_{ze} and ε :

$$\beta = \frac{0.4}{1 + 1500\varepsilon} \cdot \frac{1300}{1000 + s_{ze}} \tag{5.11}$$

We substitute β in next equation and we equal it to the applied shear:

$$V_c = \phi_c \lambda \beta \sqrt{f'_c} b_w d_v = V \tag{5.12}$$

Solving for P using Matlab we obtain $P_{nShearGeneral} = 356.22$ kN. SPLM predicts a maximum load of 379.889 kN, which is slightly higher than the Canadian Code prediction. ACI predicts a maximum shear load of $P_{nShearACI} = 596.54kN$, which is higher than the SPLM prediction and the CSA.

When the scale factor is 6, SPLM predicts a maximum load of 654.802 kN. The Canadian Code predicts a load of $P_{nBending} = 1412.901$ kN for bending and $P_{nShearSimplified} = 454.53$ kN and $P_{nShearGeneral} = 396.513$ kN for shear failure. ACI predicts a maximum shear load of $P_{nShearACI} = 910.50kN$, which again is higher than the CSA and SPLM predictions, since it does not account for size effect.

5.4 Discussion and Conclusions

SPLM is shown to reasonably predict the strength and crack pattern in reinforced concrete beams. Comparing the SPLM prediction of the strength with Canadian Code, we find that SPLM gives a slightly higher capacity. We have to remember that we can not compare directly a modelling tool with a design code, since a design code is based on experience and experimental data, and its prediction is always conservative. However, this comparison gives us an idea of where we are in the improvement process, and what we need to modify next.

Furthermore, the cracking patterns seem realistic in all of the cases, knowing that it is also affected by the bond-slip model. I think the main reason bond slip takes place in almost all the simulations is because the reinforcement has been modelled as two groups of bars. This fact makes the diameter of the reinforcement very big and therefore, the force is concentrated only in a few particles.

As mentioned in the size effect section, there is a limitation regarding computational capacity. This tests require an enormous computational capacity and that makes it unfeasible to run big tests with the resources we have available at this time. In addition, since we have to limit the duration of the tests, in some cases the loading conditions makes the problem somewhere between a quasi-static and a dynamic problem. I think it is essential to verify the results using a non-linear dynamic analysis, with Abaqus or Ansys for example, or compare the results with experimental data.

In conclusion, we have shown that SPLM is capable of modelling reinforced concrete beams with a reasonable level of accuracy. It is possible to detect and consider the limitations of the model to continue improving it.

Chapter 6

High Performance Computing

Small SPLM models can be run on a personal computer. However, for large models with hundreds of thousands of particles and millions of time steps it is necessary to run the program in a parallel computer. In this chapter we are introducing the basics of High Performance Computing (HPC) and the OpenMP and MPI parallel libraries. MPI is employed in the program pdQ, which allows SPLM to run on a massively parallel supercomputer. A hybrid OpenMP plus MPI implementation of pdQ is presented. A scalability analysis of the pure MPI system (original pdQ) and the new hybrid OpenMP plus MPI system in pdQ is included.

6.1 Basics of High Performance Computing

Flynn's Taxonomy is commonly used to classify parallel computer architectures (Pacheco, 2011). The first architecture is the Single Instruction Single Data (SISD), which is the classical Von Neumann architecture. In a Single Instruction Multiple Data (SIMD) system, the same instruction is applied to multiple data. This system can be thought as having a single control unit and multiple Arithmetic Logic Units

(ALUs). This architecture is commonly used in vector programming and Graphics Processing Unit (GPU) programming. In a Multiple Instruction Single Data (MISD) architecture, many cores work on the same data buffer. It is widely used when the communication is more expensive than the computation itself, and also with high-reliability computers, to compare results between cores. The last parallel architecture is the Multiple Instruction Multiple Data (MIMD). It typically consists of a collection of fully independent processing units or cores, each of which has its own control unit and its own ALU. Inside, each instruction can be further parallelized. This is the architecture of a "cluster", or a massive parallel computer or "supercomputer".

MIMD systems can be mainly classified into two types: shared-memory systems and distributed-memory systems. In shared-memory systems, a collection of autonomous processors are connected to a memory system via an interconnection network, and each processor can access each memory location. In this architecture, the processors usually communicate implicitly by accessing shared data structures. On the other hand, in distributed-memory systems, each processor is paired with its own private memory, and the processor-memory pairs communicate over an interconnection network. The processors usually communicate explicitly by sending messages or using special functions that provide access to the memory of another processor, as shown in Figure 6.1.

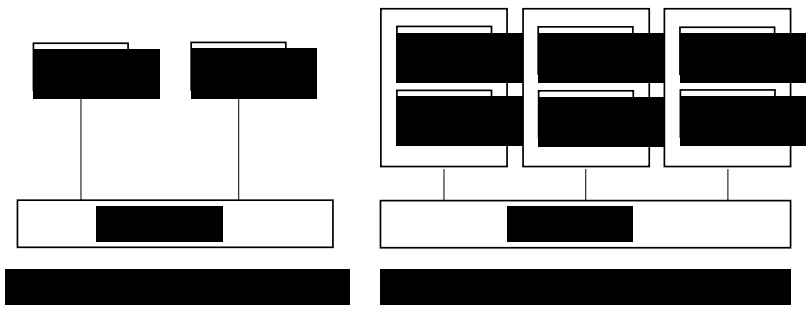


Figure 6.1: Shared-memory and distributed memory system configurations

6.2 Message Passing Interface (MPI)

MPI is the prevalent protocol for distributed-memory systems. MPI stands for Message-Passing Interface. It consists on a library of functions and instructions that can be called from C, C++, and Fortran programs. In a MIMD system, with multiple cores per node, and distributed-memory, MPI provides the set of instructions used to send and receive data from cores that are on the same node or on different nodes. Figure 6.2 shows the basic MPI setup for a cluster with two nodes and four cores in each node. The Core 0 of Node 1 is set to be the master, therefore is in charge of setting up the communication between cores on the same node (intranodal communication) and between cores that are located on other nodes (internodal communication) through the network.

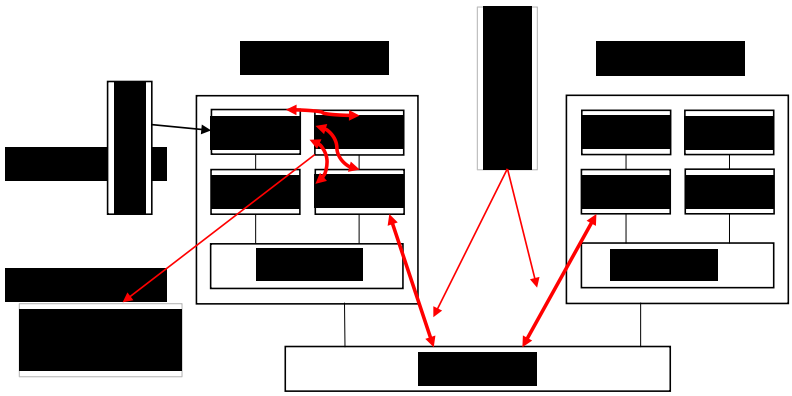


Figure 6.2: MPI setup in a MIMD system with two nodes and 4 cores per node.

The directive `MPI_INIT` must be included at the beginning of the parallel region in order for the compiler to be able to process the parallel instructions. Any MPI function called before `MPI_INIT` will result in an error. At the end of the parallel section `MPI_FINALIZE` must be invoked, so the compiler knows that from that point the code will be executed in serial, and also to free any resources allocated for MPI. The basic MPI communication is between pairs of processes in a `communicator`,

which can be `COMM_WORLD` for all the processes (it is initialized using `MPI_INIT`) or a communicator created by the user. In the following piece of code, two communication setup functions are called. `MPI_Comm_Size` returns the number of processes in the communication
`COMM_WORLD` and `MPI_Comm_Rank` the rank of all the processes inside the communicator.

```
MPI_INIT( )
MPI_Comm_Size(MPI_COMM_WORLD, &Comm_sz)
MPI_Comm_Rank(MPI_COMM_WORLD, &my_rank)
!Parallel Region
...
!End of parallel region
MPI_FINALIZE( )
```

Communication in MPI is two-sided, i.e., the sender must call `send` and `receiver` must call `receive`. In addition, the sender specifies a destination and a tag associated (an integer value), and the receiver specifies a source and a tag, and they must match. A common source of errors in MPI communication occurs when a processor calls `send` or `receive` and no processor calls `receive` or `send`. Another common issue is called deadlock. It happens when a thread is waiting for a buffer of data that nobody is sending to it, or is on a cyclic dependency (P1 is waiting for P2, P2 for P3, and so on and Pn waits for P1). In these two situations, the program will not proceed and will wait forever.

One simple way to avoid deadlocking due to communication or cycle dependency is to use the function `MPI_SENDRECV()`, in which the user defines the thread that sends and the one that receives and the subsystem takes care of the synchronization and dependencies issues. This function signature is:

```
MPI_SENDRECV(ptclSend, sendSize, MPI_DOUBLE_PRECISION
```

`,toRnk, tag, ptclRecv, maxSizeMsg, MPI_DOUBLE_PRECISION, fromRnk, tag, MPI_COMM_WORLD, status, ier)` . Chapter 3 of Pacheco (Pacheco, 2011) describes in detail the setup of this instruction.

6.3 OpenMP

OpenMP is the result of an agreement between hardware vendors and compiler developers and is considered to be an "*industry standard*": it specifies a set of compiler directives, library routines, and environment variables that can be used to specifiy shared-memory parallelism in Fortran and C/C++ programs (Hermanns, 2002). "Open" means that the standard is defined through a specification accessible to anyone and "MP" stands for Multi Processing.

OpenMP is a shared-memory parallel programming system, which means that all the processors in the node can access every memory buffer directly. Hence, it is not necessary to define a communicator, as in MPI, nor to explicitly set the buffers of data that are exchanged between threads, which results in a more user-friendly implementation. In addition, since data management is handled implicitly at runtime by the compiler the probability of having a deadlock is drastically decreased.

One of the advantages of using OpenMP is that a serial compiler can run a parallel program in serial ignoring the OpenMP directives. This is possible because the OpenMP compilers can read `!$OMP` and `!$` at the beginning of the directives as OpenMP functions, whereas a normal compiler would treat those lines as comment lines. A program can contain multiple serial and parallel regions. In Figure 6.3 there are two serial and two parallel regions. Before the parallel region, Thread 0 is in charge of creating N threads, that are then merged in the serial region after the parallel region.

In OpenMP the variables used in the parallel regions can be specified as private

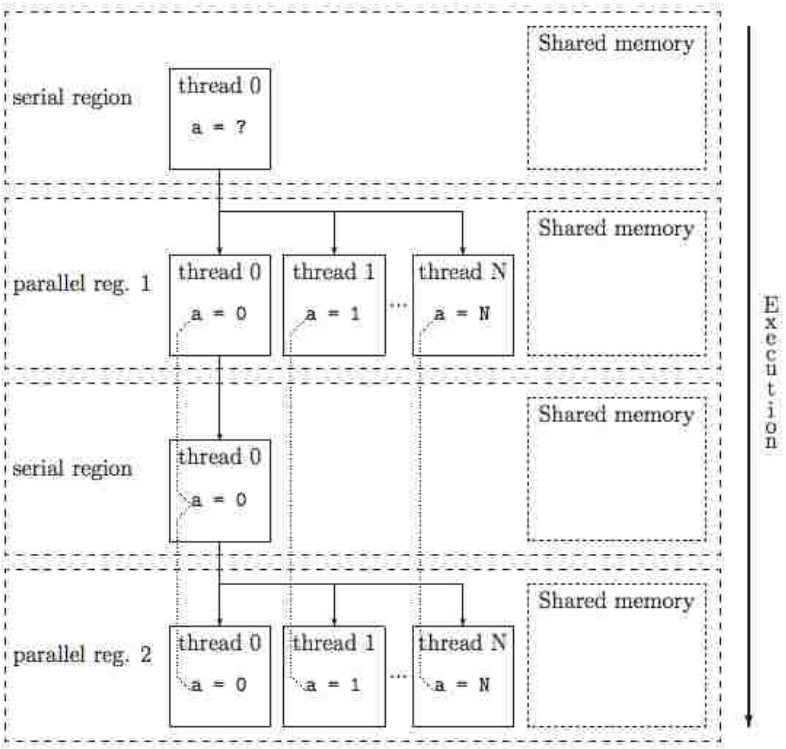


Figure 6.3: Execution of a program with openMP regions and serial regions. From (Hermanns, 2002).

or shared. When a variable is declared as private (`!$OMP PARALLEL PRIVATE(var1, var2)`) each thread has its own copy inside the parallel region, so each thread can modify it safely. Otherwise, a variable declared as shared (`!$OMP PARALLEL shared-memory(var1, var2)`) is accessible for all the threads. In Figure 6.3 the variable a is private since each thread has its own copy and a has independent values in each thread. In the case shown in Figure 6.3, the clause `!$OMP THREADPRIVATE` is invoked so each thread keep its own copy of the variable until the end of the program. For example in pdQ we must describe the particle attributes (position, velocity, force ...) as private to avoid having two or more processors overwriting the same memory location or particle attributes.

When a block of code must be executed by only one thread at a time (criti-

cal section), a function to ensure mutually exclusive access to the section is needed. If two or more processors attempt to simultaneously execute a critical section, an error can occur or the outcome of the computation may depend upon the order of the execution. This effect is called race condition. OpenMP includes the directive `!$OMP critical` to ensure that only one thread execute that block at a time. In our implementation, this function is called when the program updates the particles attributes at the end of each time step.

When a loop, that for example goes from 0 to 100, is executed by 4 threads, the master thread (usually thread 0) divides the work among the four threads, so each thread execute 25 iterations, for example. The task of assigning each thread a piece of "work" is called scheduling. In OpenMP four schedules can be specified at the beginning of the parallel region. In a static schedule, the iterations are assigned before the section is executed. On the other hand, in a dynamic schedule, the iterations are assigned to the threads while the section is executing, thus after a thread completes its current set of iterations it can request more from the run-time system. This schedule is convenient when the thread load is not balanced. OpenMP includes an automatic schedule called auto, where the compiler and/or the run-time system determine the schedule. Finally, in a runtime schedule the schedule is determined at run-time.

Although the objective of the parallelisation is to run the maximum percentage of the program in parallel, there are still some parts that need to run in serial. To delimit the parallel region, the next two directives must be called:

```
!$OMP PARALLEL clause1 clause2 ...
!Parallel Region
...
!End of parallel region
!$OMP END PARALLEL
```

In the clauses we can specify the scope of the variables (private or shared) or the schedule (static, dynamic ..).

6.4 Hybrid Implementation OpenMP + MPI in pdQ

The motivation for implementing the hybrid OpenMP and MPI is to make the most of the parallel computer resources. A cluster can be thought as a set of independent computers connected by an interconnection network. Hence, each node has its own memory, which access faster than a memory buffer located in a different node. Since each node usually has a number of processors, depending on its architecture, it is logical to think that may be feasible to implement OpenMP (shared memory) inside each node and MPI (distributed memory) between the nodes to communicate the data. With this implementation, when a number of threads, located in the same node, are working on the same data, there is no need for the processors to get their own copy of the data, unless the variables are private. Furthermore, MPI, which in some cases can be expensive, is limited to internodal communication.

Figure 6.4 shows the schematic of this implementation.

6.4.1 pdQ

PdQ is the acronym for "particle dynamics - quantum". This domain decomposition algorithm was developed by Dr. Atlas and Dr. Gerstle at the University of New Mexico, along with their graduate students Navid Sakhavand, Vijay Janardhanam, and Hossein Honarvar (Sakhavand, 2011), (Honarvar, 2013) in 2009. PdQ allows the use of parallel computers for solving SPLM problems.

PdQ decomposes the specimen into a number of subdomains defined by the

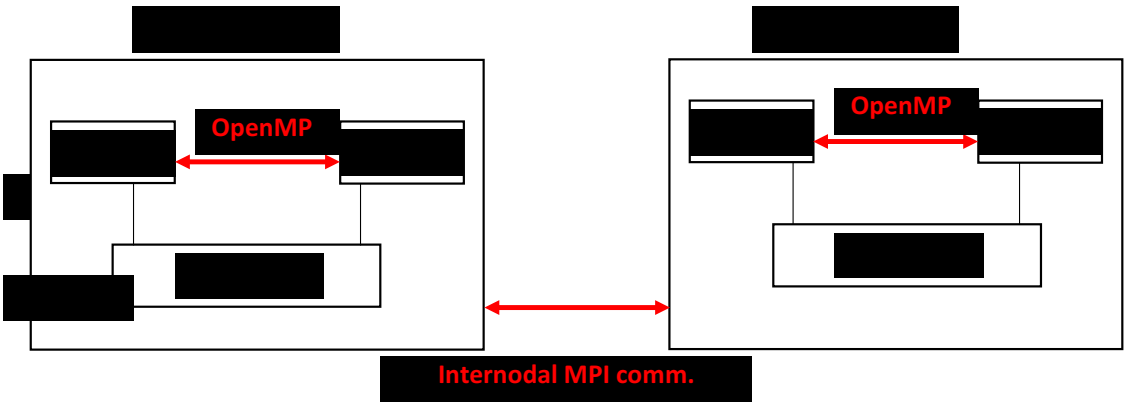


Figure 6.4: Hybrid OpenMP + MPI setup in a cluster

user. Each subdomain is assigned to a single core, and is additionally decomposed in a core and a set of walls. pdQ incorporates a set of subroutines, that set the communication for exchanging the particles that are in the walls, and at the end of a time step pull the data necessary to write to disk, when is set by the user. Figure 6.5 shows the topology of the domain decomposition. In addition, each subdomain core *iCore*, the particles are divided into cells. For further information see (Gerstle, 2015) and (Honarvar, 2013).

In each time, step after calculating the forces, displacements and mechanical properties (particle attributes) for all the particles, there is a call to a function called *ExchangePtclAttributes*, that sets the communication between cores for the particles that are in the same wall for two different processors. Depending on the problem size this function doesn't usually take more than a 20% of the total processing time, and above this percentage it is recommended to decrease the number of processors to boost the efficiency.

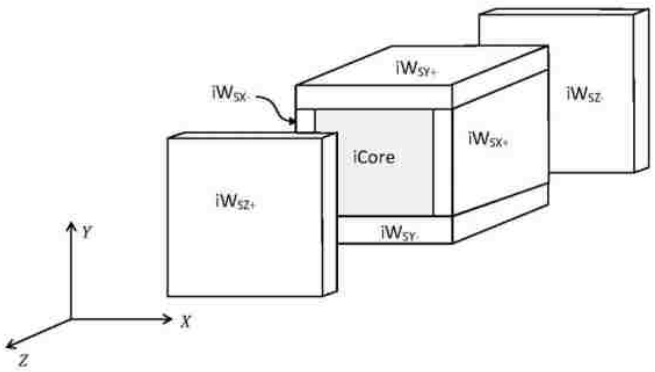


Figure 13.2. The home procCube, its core, and its sending walls.

Figure 6.5: Particle exchange between processors. (From (Gerstle, 2015)).

6.4.2 OpenMP implementation in userForce

The force calculation subroutine usually takes around 70% of the total time. This subroutine loops through all the particles of the subdomain, computing the state forces and stretches. In the domain decomposition, the body is divided in the number of MPI cores and in addition, each subdomain is divided in cells in X, Y and Z to accelerate the calculations. The cell decomposition makes use of the neighbor matrix ($\mathcal{N}[P_i]\langle B_j\rangle$). Thus, each particle need only to loop over its neighbor list. Consequently, openMP can be easily implemented in the subroutine `userforce`, and the total number of cells in the domain are divided among the openMP threads.

In this implementation if for example we have a specimen with 1000 particles and we can use two nodes of the cluster, each node is having 500 particles. These 500 particles are also divided into 100 cells, for example. If each node has 8 processors, in each time step when the program executes the module `userForce` 8 openMP threads are initiated. Thus, each openMP thread works on 12 cells (integer(100/8)). When all the threads have finished, the openMP master thread close them. The program continues to run in serial, but using just by one MPI thread on each node. At the

end of each time step, the master MPI thread at each node sends and receives the data for the particles in the wall.

6.5 Scalability Analysis

To measure the wall time of the program, pdQ outputs a text document which contains the `MPI_WALL` time in each subroutine (userforce, writeout, exchangeptclattrs ...) and the percentage of the total time. A simulation constant, **K**, introduced by Gerstle (Gerstle, 2015), is also output to provide a unique measure of the performance of the code. K is defined as the ratio $\frac{processor-seconds}{particle-timestep}$ and is usually around $2 \cdot 10^{-5}$ $\frac{processor-seconds}{particle-timestep}$ for a parallel computer of the 2010 era.

To study the scalability of the original pdQ and the hybrid OpenMP plus MPI implementation, a strong scaling test is performed. This test consists of measuring the time spent by the program, for a large fixed size, while increasing the number of processors. This test was repeated five times using 1, 2, 4, 8 and 16 processors, and 8 processors per node.

Figure 6.6 shows the total running time spent in the main subroutines. Some subroutines (CellAndProcDecomp, Populate, userSetup, userWriteOut and UserIntegrate) are only executed by one core. For codes displaying linear scaling, when increasing from 1 to 2 the number of processors the total time should decrease by a factor of 2. However, Figure 6.6 shows an increase in time when two processors are used. Looking at the figure one can see that the time spent in the subroutines in exchange of particles and write out, have significantly increased, while the time spent in other subroutines has not decreased in the same rate. This shows that for problems with a half-million particles, the communication and write out time dominates over the computation time. Figure 6.7 presents a closer view of the subroutine userForce. The parallel ideal time T_{ideal} is defined as the serial time, T_{serial} over the

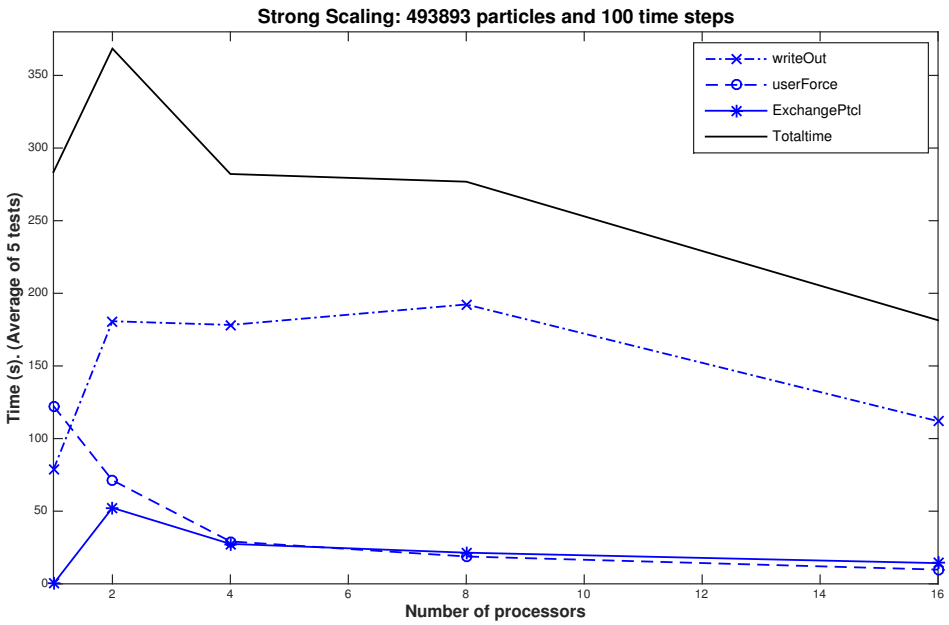


Figure 6.6: Strong Scaling test in the original pdQ

number of processors p, $T_ideal = T_{serial}/p$. When comparing the "real" time with T_{ideal}, we see that the `userForce` subroutine strongly scales.

Figure 6.8 contains the wall times for four subroutines: total time, `ExchangePtcls`, `cell and processor decomposition` and `writeOut`. The total time curve presents a peak, when 2 processors are running, due to the increase of time in exchange particles and write out subroutines. The `ExchangePtcls` subroutine shows a decrease of time when increasing the number of processors, since the data buffers are smaller. The subroutine domain decomposition (cellAndProc) stays constant, as expected, since the domain decomposition is done by one processor. With these results we can conclude that we need to do the write out to file more efficiently in order to be able to simulate and postprocess larger problems.

Figure 6.9 contains the strong scaling results for the hybrid pdQ. The total

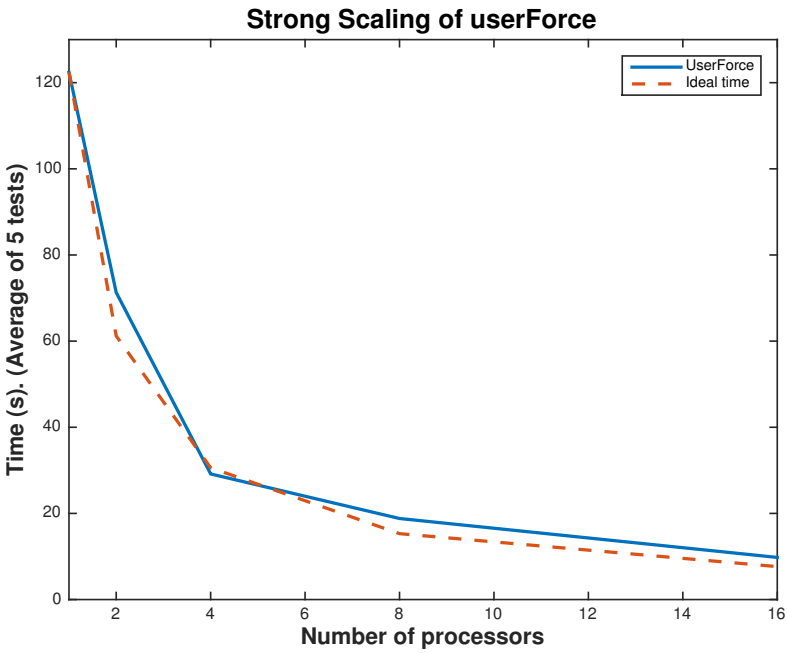


Figure 6.7: Strong Scaling for userForce in the original pdQ

time increases linearly with the number of threads, reaching its maximum when 8 processors are running. When 16 processors are running, two nodes are working, and both openMP (intranodal communication) and MPI (internodal communication) are active. Hence, the total time should increase because of the communication between nodes, but the response is the opposite.

Figure 6.10 shows the strong scaling test results for both pdQ versions. While the pure MPI code (original pdQ) somehow scales, the hybrid system response seems completely wrong. This issue has been reported in the literature (Rabenseifner et al., 2013). Against all predictions, the pure MPI implementation is more efficient than the hybrid. The key factor in this problem is the cache memory.

The cache memory is a collection of memory locations, that are accessed faster than other memory locations (Pacheco, 2011). It is divided into levels: the first level (L1) is the smallest and the fastest, and higher levels (L2, L3, ..) are larger and

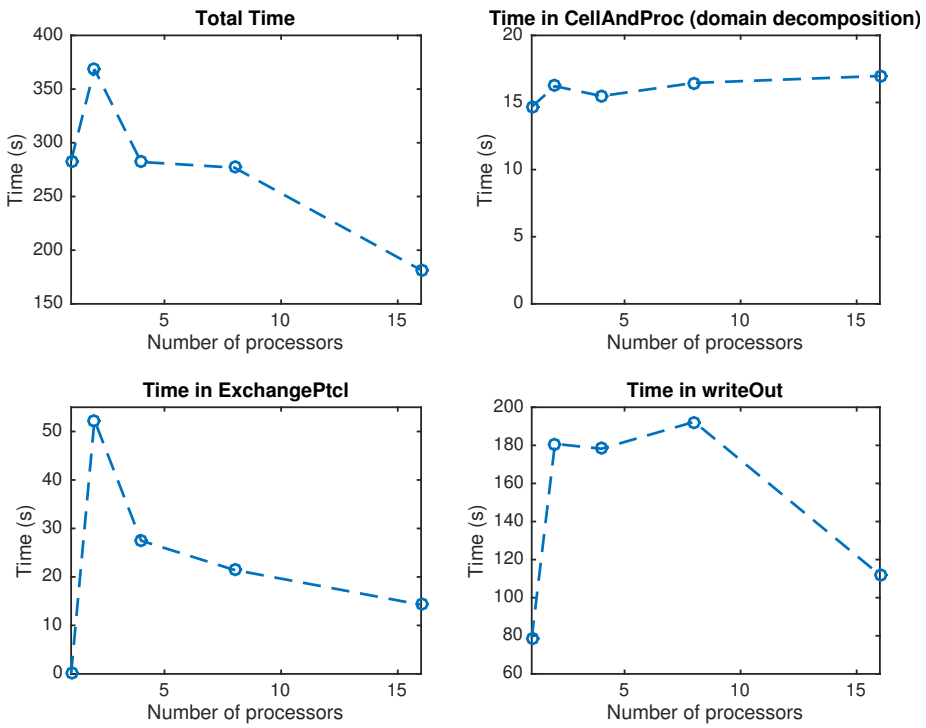


Figure 6.8: Strong Scaling of 4 subroutines in the original pdQ

slower. In practice when the CPU needs to access and instruction or data it starts from the first level and goes down hierarchy. In MPI, the variables that are used by each processor at each time are explicitly defined by the programmer. Hence, when a processor needs a variable, it is copied to cache memory, and it remains there until the same processor needs another variable, so it is copied to the cache memory. The fewer times a processor needs to copy a data buffer to the cache memory, the more efficient is the algorithm. On the other hand, data treatment in OpenMP is implicitly defined by the compiler, consequently, the programmer is not able to define the data buffer copies to cache memory. Hence, the increased in time showed in Figure 6.10 is due to all the processors copying and deleting data from the cache memory.

Another source of performance issues is the pdQ data structure. When the element of an array is needed by a processor, the compiler loads it into the cache

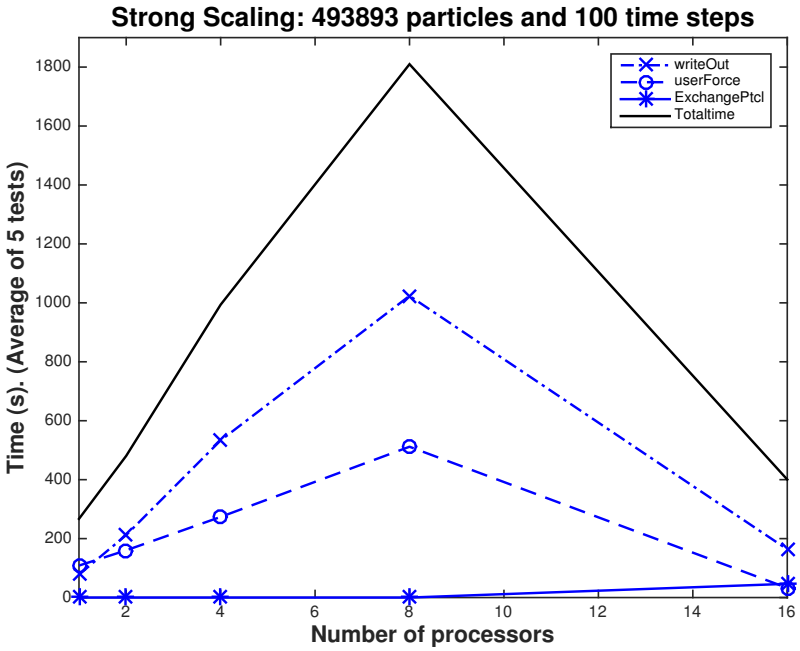


Figure 6.9: Strong scaling of the hybrid pdQ

memory first. But it does not load just one element of the array. Instead, a piece of the array is loaded, which size and orientation depends on the compiler. In C and C++, the compiler loads a row of elements, while the Fortran compiler loads a column. The `userForce` subroutine loops through all the particles in the domain within each time step. The particle attributes are loaded in a row of elements. As a result, in a single time step the compiler needs to copy the same data buffer multiple times.

An attempt was made to change the data structure. However, due to lack of time it was not successful. This alteration of the pdQ code is therefore proposed as future research.

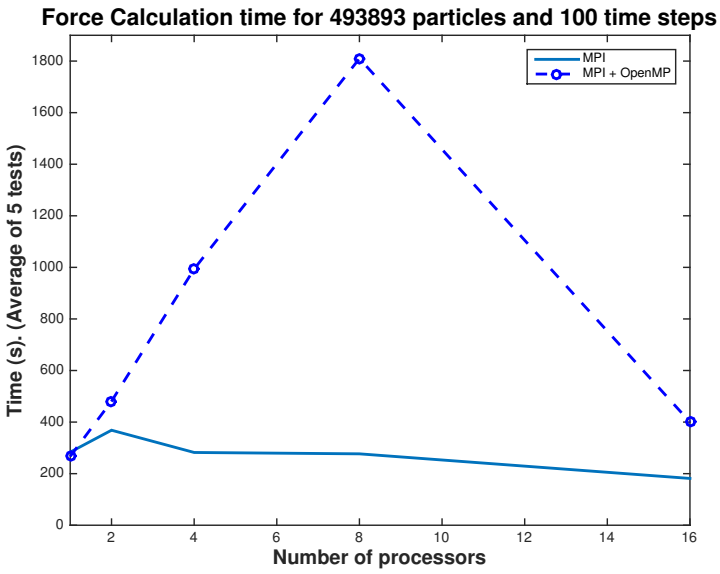


Figure 6.10: Strong Scaling test for userforce subroutine for original pdQ and hybrid pdQ.

6.6 Conclusions

In conclusion, the original pdQ is shown to be not scalable. The subroutine `userForce` in the original pdQ is strongly scalable. We can conclude that we should work to improve the communication between processors and the write out algorithm. The hybrid implementation presented performed worse than the original pdQ. The reason for this poor performance is the data treatment by OpenMP and the trashing of the cache memory. As future work, the data structure in pdQ should be changed, in order to be more efficient. In addition, as shown in previous chapters, computational capacity is still a limitation on the size and duration of the simulations. Perhaps pdQ is more suitable for a different parallel computer architecture, with millions of processors, thus each processor has only one particle. But we still need to find more efficient ways to run pdQ with the parallel computers that we have now.

Chapter 7

Conclusions

7.1 Summary

In this thesis we presented a new constitutive model for the State-Based Peridynamic Lattice Model (SPLM), applied to concrete structures. We started with a review of some continuum models, the elastic SPLM model and the simplified modified compression field theory.

We developed the bond peridynamic lattice damage model, which we found not to be insufficiently objective to model concrete structures. This finding justified the development of the SPLM damage model, which is more general. We verified the model using uniaxial tension and uniaxial compression tests.

We continued working on the new constitutive model by studying and modifying the plasticity model. We tested the model using the Brazilian split cylinder test. With the new constitutive model completely defined (elasticity, plasticity, and damage) we compared it with some continuum finite element models.

We also applied the new SPLM to reinforced concrete beams. We performed some simulations where we could study the effect of the longitudinal reinforcement,

the bond-slip of the rebars, and the size effect of the beams. We used the Canadian and ACI codes for comparison.

Finally, we presented some basics of high performance computing. We developed a new parallel implementation of pdQ using OpenMP plus MPI, and studied the scalability of the original and new implementations.

7.2 Major Findings

We have found that the bond peridyamic lattice damage model (BPLDM) is not objective with respect to lattice rotation. We conclude that a bond-based approach is inadequate for modelling damage in peridynamics. We found that the SPLM damage model gives a more objective prediction, but is still slightly dependent on lattice rotation.

We also found that the SPLM plasticity model rationally simulates plastic flow in a concrete structure. However, the strength prediction for the Brazilian split cylinder is not close from the literature. When we applied the new SPLM plasticity and damage models to reinforced concrete beams, we found that both the maximum and load cracking patterns are reasonably simulated. We showed then, that SPLM is capable of modelling plain concrete structures and reinforced concrete structures.

Finally, we found that an implementation of openMP plus MPI in our code is less efficient than the original pure MPI implementation. We also found that the current data structure is not optimized for the Fortran compiler, and the write out is not efficient.

7.3 Future Research

If we want SPLM to be a viable alternative to continuum models for design engineering, we need to make some improvements.

The SPLM damage model has only been tested under plane stress fields. We need to explore the response under different conditions. In addition, the stiffness degradation function is computed in terms of the average stretch at a particle. We need to investigate other options, like being a function of the maximum stretch, or the stretch in the bond direction.

The SPLM plasticity model is predicting a reasonable response, but for the splitting strength of the Brazilian split cylinder, that we need to revisit. The second invariant J_2 computed from the average force in each bond direction, is perhaps too simple. Hence, we should explore how the prediction changes when J_2 is calculated without taking the average force in the bonds. In addition, the damage due to excessive plastic flow is oversimplified, and a more gradual model should be developed. It may be interesting to develop a plasticity model whose yield surface evolve in time, to account for hardening or softening phenomena.

In the SPLM reinforced concrete beam study, it was recognized that in some cases, the failure mechanism of the beam was bond-slip. We need to refine the bond-slip model. The steel area is divided into a number of bars in the vertical direction, usually one or two. We should divide the steel area in the vertical and horizontal, so the steel area is distributed along the beam section better.

Finally, it is essential to overcome the computational limitations, such as the data structure of the code, communication or write out. We need to improve the code to make it more efficient and explore some alternative parallel implementations.

Bibliography

[1] ACI318. (2008). *ACI 318 Building Code Requirements for Structural Concrete.* Farmington Hills, Michigan: American Concrete Institute.

[2] ACI446 (2004). *ACI 446.4R-04: Report on dynamic fracture of concrete.* Farmington Hills, Michigan: American Concrete Institute.

[3] ACI446 (2009). *ACI 446.3R-97: Finite element analysis of fracture in concrete structures: state-of-the-art.* Farmington Hills, Michigan: American Concrete Institute.

[4] Bazant, Z. P. and Planas, J. P. (1998). *Fracture and size effect in concrete and other quasibrittle materials.* Boca Raton, Florida: CRC Press.

[5] Bazant, Z. P., and Yu, Q. (2004). *Size effect in concrete specimens and structures: new problems and progress.* Fracture Mechanics of Concrete Structures, 153-162.

[6] Bentz, E. V., Vecchio, F. J., and Collins, M. P. (2006). *Simplified Modified Compression Field Theory for Calculating Shear Strength of Reinforced Concrete Elements.* ACI Structural Journal, July-August 2006, Tittle no. 103-S65.

[7] Canadian Standard Association (2004). *A23.3 - 04. Design of Concrete Structures.* Ottawa: Canadian Standards Association.

[8] Carmona, S., Gettu, R., and Aguado, A. (1998). *Study of the post-peak behavior of concrete in the splitting-tension test.* Proceedings FRAMCOS-3, AEDIFICATIO, Freiburg, Germany.

[9] Chapman, S. (2011). *MS Thesis: Clarification of the notched beam level II testing procedures of ACI 446 committee report 5* Albuquerque: University of New Mexico.

[10] Gerstle, W. H. (2015). *Introduction to practical peridynamics: Computational solid mechanics without stress and strain.* (Frontier research in computation and mechanics of materials and biology, vol. 1). New Jersey: World Scientific.

[11] Hermanns, M., (2002). *Parallel Programming in Fortran 95 using OpenMP.* Universidad Politecnica de Madrid, April, 2002, Spain.

[12] Hillerborg, A., Modeer, M., and Petersson, P. E. (1976). *Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements.* Cement and Concrete Research, 773-782.

[13] Hillerborg, A. (1978). *A model for fracture analysis.* (Report TVBM; Vol. 3005). Division of Building Materials, LTH, Lund University.

[14] Honarvar Gheitanbaf, H., (2013). *MS Thesis: Parallel simulation of particle dynamics with application to micropolar peridynamic lattice modeling of reinforced concrete Structures.* Albuquerque: University of New Mexico.

[15] Lee, J., and Fenves, G. L. (1998). *Plastic-Damage model for cyclic loading of concrete structures.* Journal of Engineering Mechanics: 124(8): 892-900.

[16] McVey, S. M. (2015). *MS Thesis: The state-based peridynamic lattice model and reinforced concrete structures.* Albuquerque: The University of New Mexico.

[17] Lubliner, J., Oliver, J., Oller, S., and Onate, E. (1989). *A plastic-damage model for concrete.* International Journal of Solids and Structures: January 1989, Vol. 25, No. 3, pp 299-326.

[18] Lubliner, J. (2008). *Plasticity Theory.* New York: Courier Corporation.

[19] Malarics, V., and Muller, H. S. (2010). *Evaluation of splitting tension test for concrete from a fracture mechanical point of view.* Proceedings of fraMCoS-7, May 23-28, 2010.

[20] Pacheco, P. S., (2011). *An Introduction to Parallel Programming.* Morgan Kaufmann, 2011, USA.

[21] Rabenseifner, R., Hager, G., Jost, G., (2013). *Hybrid MPI and OpenMP. Parallel Programming.* Tutorial at SC13, November 17, 2013, Denver (CO) USA.

[22] Silling, S.A. (2000). *Reformulation of elasticity theory for discontinuities and long-range forces.* Journal of the Mechanics and Physics of Solids: 2000, V. 48, 175-209.

[23] Silling, S.A., Epton, M., Weckner, O., Xu, J., and Askari, E. (2007). *Peridynamic States and Constitutive Modeling.* Journal of Elasticity, pp. 151-184.

[24] Simulia (2015). *Abaqus 6.14 user's guide.* Simulia (http://bobcat.nus.edu.sg:2080/v6.14/index.html).

[25] Sakhavand, N., (2011). *MS Thesis: Parallel Simulation of Reinforced Concrete Structures Using Peridynamics.* Albuquerque: University of New Mexico.

[26] Vecchio, F. J., and Collins, M. P. (1986). *The Modified Compression-Field Theory for Reinforced Concrete Elements Subjected to Shear.* ACI Journal, March-April, tittle no. 83-22.

[27] Vemuganti, S. (2015). *MS Thesis: Analysis of brazilian split cylinder using the state based peridynamic lattice model.* Albuquerque: University of New Mexico.

[28] Witmann, F. H., Rokugo, K., Bruhwiler, E., Mihashi, H., and Simonin, P. (1988). *Fracture energy and strain softening of concrete as determined by means of compact tension specimens.* Materials and Structures: 1988, 21, 21-32.

[29] Wright, P. J. F. (1955). *Comments on an indirect tensile test on concrete cylinders.* Magazine of Concrete Resarch, 7, 20, 87-95.