



ÉCOLE POLYTECHNIQUE  
FÉDÉRALE DE LAUSANNE

MASTER THESIS

Laboratoire de simulation en mécanique des solides

---

# Developing a 3D finite element model for reinforced concrete

---

*Author:*

Lucas FRÉROT

*Supervisors:*

Mr. Marco VOCIALTA

Dr. Mauro CORRADO

Dr. Thomas MENOUEILLARD

Prof. Jean-François MOLINARI

July 21, 2015

# Summary

The embedded model is used in finite elements to account for the influence of the reinforcement in concrete. This report describes the embedded model, its concept and equations. Details are also given as to the constitutive relations of concrete and steel. An algorithm is described for an implementation of the embedded model in a finite element code (in this paper, the open source library Akantu is used). The implementation is done in 2D and 3D, includes the computation of a stiffness matrix, a residual vector, allows for prestressing of the reinforcements and for nonlinearities by swapping of the constitutive laws. Results given by the embedded model are compared to those given by another model used to represent reinforcements in concrete : the discrete model. The convergence of both models are analyzed in terms of potential energy and stress norm, although some issues arise with the latter. The embedded model is tested with a real-sized structure, a prestressed concrete bridge for which the fatigue solicitation is estimated, but also with nonlinear constitutive laws for both concrete and reinforcements.

# Contents

<b>1</b>	<b>Modeling reinforced concrete</b>	<b>5</b>
1.1	Modeling reinforcement . . . . .	5
1.2	Reinforcement behavior . . . . .	10
1.2.1	Passive reinforcements . . . . .	10
1.2.2	Prestressing cables . . . . .	11
1.3	Concrete behavior . . . . .	11
<b>2</b>	<b>Embedded model, method and algorithm</b>	<b>13</b>
2.1	Computing intersections . . . . .	14
2.2	Stiffness matrix . . . . .	15
2.3	Residual forces . . . . .	15
<b>3</b>	<b>Comparison with discrete model</b>	<b>17</b>
3.1	Comparison example . . . . .	17
3.2	Convergence analysis . . . . .	20
<b>4</b>	<b>Prestressing</b>	<b>26</b>
4.1	Effect of prestressing . . . . .	26
4.2	Numerical application . . . . .	27
4.2.1	Analytical solutions . . . . .	28
4.2.2	Convergence analysis . . . . .	31
4.3	Stress distribution in reinforcement . . . . .	33
<b>5</b>	<b>Application to concrete bridge</b>	<b>37</b>
5.1	Description of the bridge . . . . .	37
5.1.1	Geometrical and material characteristics . . . . .	37
5.1.2	Finite elements characteristics . . . . .	40
5.2	Fatigue solicitation . . . . .	41
5.3	Results . . . . .	41
5.3.1	Dead loads . . . . .	41
5.3.2	Fatigue . . . . .	42
<b>6</b>	<b>Preliminary nonlinear analysis</b>	<b>44</b>
6.1	Model description . . . . .	44
6.2	Result analysis . . . . .	46
6.2.1	Force displacement behavior . . . . .	46
6.2.2	Damage propagation and cracks . . . . .	50

<b>A</b>	<b>Computational geometry, code organization and algorithms</b>	<b>54</b>
A.1	Embedded model and materials . . . . .	54
A.2	Mesh geometry module . . . . .	55
<b>B</b>	<b>Constitutive laws and algorithms</b>	<b>57</b>
B.1	Isotropic linear hardening . . . . .	57
B.2	Regularized sequentially linear saw-tooth softening . . . . .	59

# Introduction

Numerical modeling is becoming an increasingly more important part of a civil engineer's work. As our wishes for more complex and original structures grow, an engineer must expand their means of design and computation of structures. Thus comes the need for precise modeling, often done with the finite element method. This method has been used for decades in a variety of different fields of the civil engineering domain : buildings, dams, bridges, tunnels, etc., and it has become the method of choice for numerical modeling.

Many have adapted the finite element method to model the behavior of materials used in civil engineering. In this project, we will focus on the modeling of reinforced concrete. This subject is centered around two problems that need to be solved in order to successfully model the behavior of reinforced concrete :

- How to model the steel reinforcement embedded in concrete ?
- How to model the concrete behavior ?

These questions will be answered in the first part of this thesis (chapters 1 through 3), with an emphasis put on the first question. This work was not done during the semester dedicated to the master project, but during the master pre-study, a semester earlier. It is included in this report for the convenience of the reader. The work done during the master project proper focuses on three distinct subjects :

- Conformity of the steel reinforcement model with analytical solutions
- A real life application of the developed model where hand calculations become tedious
- An application of the model with non-linear constitutive laws for both the concrete and the reinforcements

# Chapter 1

## Modeling reinforced concrete

Modeling reinforced concrete behavior is a challenge. Reinforced concrete is a composite, heterogeneous, anisotropic and non-linear material. Simplified models, that an engineer can use, exist to allow efficient design of a structure. However, many models used by engineers fail to predict the response of a concrete structure in terms of displacements with sufficient accuracy. This requires a non-linear analysis of the structure, which becomes tedious if done by hand. Furthermore, structures are becoming more complex, increasing the need for numerical simulations. With reinforced concrete, elastic models are not enough to represent the behavior of the material, and this can lead to errors in the predicted response of the structure.

To improve the quality of numerical models for finite elements, several methods have been developed to account for the following aspects of concrete :

1. Reinforcement layout (position and quantity)
2. Reinforcement behavior (plasticity, but also pre-/post-tensioning effects)
3. Concrete behavior (plasticity, damage and delayed effects)

In this part of the thesis, we will mainly focus on point 1, and briefly explain the material behavior.

### 1.1 Modeling reinforcement

There are several methods of accounting for the reinforcement in a reinforced concrete finite elements model. The following are the three main models [8] :

1. Smeared model
2. Discrete model
3. Embedded model

The first model assumes a distributed amount of reinforcement in an element. As mentioned in [8], “the constitutive relations can be derived from the composite theory.” This means that the constitutive tensor  $\mathbf{D}$  can be expressed as a weighted sum of the

constitutive tensor of the concrete  $\mathbf{D}_c$  and the constitutive tensor of the reinforcement  $\mathbf{D}_s$ , as shown in [8] :

$$\mathbf{D} = \rho_c \mathbf{D}_c + \rho_s \mathbf{D}_s \quad (1.1)$$

Where  $\rho_s$  and  $\rho_c$  are volume ratios of steel and concrete in one element. This model works well for distributed reinforcements, which is typically the case of slab reinforcements. However, it does not model the reinforcements explicitly, and thus fails to accurately model concrete elements with discrete reinforcements.

The second model can be used to handle those cases where the smeared model falls short. The discrete model represents each reinforcement bar explicitly as a spring between two nodes of the mesh. Instead of modifying the constitutive tensor of the elements of the mesh, it acts on the global stiffness matrix :

$$\mathbf{K} = \mathbf{K}_c + \sum_{i=1}^N \mathbf{K}_{s,i} \quad (1.2)$$

Where  $\mathbf{K}_c$  is the global concrete stiffness matrix computed as  $\mathbf{K}_c = \int_{\Omega} \mathbf{B}^T \mathbf{D}_c \mathbf{B} dV$  (derived from finite elements theory [11]), and  $\mathbf{K}_{s,i}$  is the stiffness matrix of the  $i$ -th reinforcement, which can be expressed as (in 2D, elastic linear, local form) [11] :

$$\mathbf{K}_{s,i} = \begin{pmatrix} \frac{E_i A_i}{L_i} & 0 & -\frac{E_i A_i}{L_i} & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{E_i A_i}{L_i} & 0 & \frac{E_i A_i}{L_i} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{in principal axes} \quad (1.3)$$

This model allows for an accurate behavior of the reinforcement, especially for structures with discrete rebars, assuming a perfect bond between the steel and the concrete. But it has the drawback that one needs to mesh the solid according to the reinforcement layout. This can be a minor inconvenience in the case of beams with standard reinforcing cages, as the reinforcement layout is regular and orthogonal, and fits a mesh of quadrangle elements. However, it can become a problem when that reinforcement layout is coupled with post-tension cables, which typically have parabolic shapes. The mesh produced can have badly shaped elements, and the computation loses in accuracy. Adapting the mesh to the reinforcement layout can also lead to smaller than necessary elements, increasing the computation time. In order to avoid that, the user has to merge two or more rebars into one. Furthermore, meshing a 3D solid according to a reinforcement layout can become very complicated [13].

The third model, the embedded model, allows the reinforcement bars to go through the mesh, but not necessarily through the nodes. The reinforcements are still modeled explicitly, but unlike the previous model, the modification of the stiffness matrix is done on a local (element) scale :

$$\mathbf{K}_e = \mathbf{K}_c + \sum_{i=1}^R \mathbf{K}_{s,i} = \int_{\Omega_e} \mathbf{B}^T \mathbf{D}_c \mathbf{B} dV + \sum_{i=1}^R A_i \int_{S_i} \mathbf{B}^T \mathbf{C}_i^T \mathbf{D}_{s,i} \mathbf{C}_i \mathbf{B} ds \quad (1.4)$$

For element  $e$ ,  $R$  is the number of reinforcements crossing  $e$ ,  $S_i$  is the length of the  $i$ -th reinforcement within  $e$ ,  $\mathbf{C}_i$  is the matrix that links the reinforcement strains to the

concrete strains. By assuming a perfect bond between the steel and the concrete, the axial strain of the bar can be expressed by equation (1.5). Then  $\mathbf{C}_i$  can be derived from the Voigt notation of the strains, as shown in equation (1.6) [9] for the 3D case :

$$\varepsilon_{s,x} = l^2 \varepsilon_{c,x} + m^2 \varepsilon_{c,y} + n^2 \varepsilon_{c,z} + lm \gamma_{c,xy} + ln \gamma_{c,yz} + mn \gamma_{c,xz} \quad (1.5)$$

$$\mathbf{C} = \begin{pmatrix} l^2 & m^2 & n^2 & lm & ln & mn \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.6)$$

In those equations,  $l$ ,  $m$  and  $n$  are the directing cosines. Let  $\vec{r}(t)$  be a parametrization of the  $i$ -th reinforcement. Then the directing cosines are :

$$l = \left\| \frac{d\vec{r}}{dt} \right\|^{-1} \left( \frac{d\vec{r}}{dt} \right)_x \quad (1.7)$$

$$m = \left\| \frac{d\vec{r}}{dt} \right\|^{-1} \left( \frac{d\vec{r}}{dt} \right)_y \quad (1.8)$$

$$n = \left\| \frac{d\vec{r}}{dt} \right\|^{-1} \left( \frac{d\vec{r}}{dt} \right)_z \quad (1.9)$$

Finally, in equation (1.4),  $\mathbf{B}$  is the matrix that contains the derivatives of the shape functions related to  $e$ , and  $A_i$  is the cross-sectional area of the  $i$ -th reinforcement.

The embedded model has two advantages over the discrete model : it allows for an arbitrary mesh of the concrete, and can represent the behavior of reinforcements with higher order shapes (like a parabola), which is the case for post-tension cables, without having to discretize the cable in a piecewise linear parametrization.

Even though the mesh can be completely independent of the reinforcement layout, reinforcements still represent singularities in the solid, especially at the end of the bar, which needs to be anchored in the concrete, creating a stress concentration. It is thus wise to refine the mesh at the end of a bar, or at any other singularity that reinforcements could be creating.

It can be shown that for the case where the reinforcement would cross an element along one of its edges, thus passing through the nodes of the mesh, the embedded model is equivalent to the discrete model, meaning that the stiffness matrix that is added to the concrete stiffness matrix is the same in both models. This means that the embedded model is a generalization of the discrete model.

It is interesting to note the differences between the two models on an element scale. Let us compare two cases illustrated in figure 1.1.



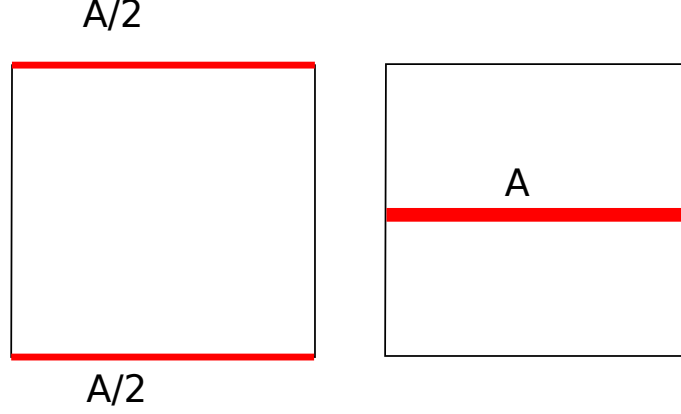


Figure 1.1: Two square elements of dimension  $2 \times 2$ . Left has two reinforcements with area  $\frac{A}{2}$  each. Right has one reinforcement at mid-height, with area  $A$ .

Only the matrices  $\mathbf{K}_{s,i}$  on an element basis are computed, since the concrete stiffness matrices  $\mathbf{K}_c$  would be the same in both cases. The stiffness matrix for the left case can be computed with both the embedded and the discrete models. For the right case, the stiffness matrix needs to be computed with equation (1.10) :

$$\mathbf{K}_{s,i} = A_i \int_{S_i} \mathbf{B}^T \mathbf{C}_i^T \mathbf{D}_{s,i} \mathbf{C}_i \mathbf{B} \, ds \quad (1.10)$$

Considering the orthogonal  $(x, y)$  basis, with  $x$  aligned with the reinforcement, the directions cosines are  $l = 1$ ,  $m = 0$  and  $n = 0$  :

$$\mathbf{C}_i^T \mathbf{D}_{s,i} \mathbf{C}_i = \begin{pmatrix} E & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (1.11)$$

From the finite element theory, the expression of  $\mathbf{B}$  for a quadrangle element with order 1 shape functions is, in Voigt notation [2] :

$$\mathbf{B} = \begin{pmatrix} -\frac{1}{4}(1-y) & 0 & \frac{1}{4}(1-y) & 0 & \frac{1}{4}(1+y) & 0 & -\frac{1}{4}(1+y) & 0 \\ 0 & -\frac{1}{4}(1-x) & 0 & -\frac{1}{4}(1+x) & 0 & \frac{1}{4}(1+x) & 0 & \frac{1}{4}(1-x) \\ -\frac{1}{4}(1-x) & -\frac{1}{4}(1-y) & -\frac{1}{4}(1+x) & \frac{1}{4}(1-y) & \frac{1}{4}(1+x) & \frac{1}{4}(1+y) & \frac{1}{4}(1-x) & -\frac{1}{4}(1+y) \end{pmatrix} \quad (1.12)$$

The computation of equation (1.10) gives  $\mathbf{K}_{s,i}$  as a function of the position  $y$  of the reinforcement, with respect to the center of the element (for the left case,  $y \in \{1, -1\}$ , for the right case,  $y = 0$ , as shown in figure 1.2) :

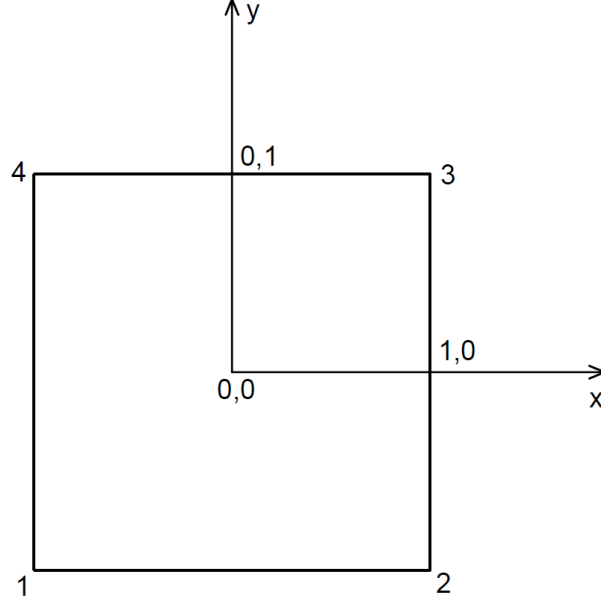


Figure 1.2: Element numbering and node coordinates

$$\mathbf{K}_{s,i} = \begin{pmatrix} \frac{EA}{8}(1-y)^2 & 0 & -\frac{EA}{8}(1-y)^2 & 0 & -\frac{EA}{8}(1+y)(1-y) & 0 & \frac{EA}{8}(1+y)(1-y) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{8}(1-y)^2 & 0 & \frac{EA}{8}(1-y)^2 & 0 & \frac{EA}{8}(1+y)(1-y) & 0 & -\frac{EA}{8}(1+y)(1-y) & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{EA}{8}(1+y)(1-y) & 0 & -\frac{EA}{8}(1+y)(1-y) & 0 & \frac{EA}{8}(1+y)^2 & 0 & -\frac{EA}{8}(1+y)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{8}(1+y)(1-y) & 0 & \frac{EA}{8}(1+y)(1-y) & 0 & -\frac{EA}{8}(1+y)^2 & 0 & \frac{EA}{8}(1+y)^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.13)$$

The assembled matrices  $\mathbf{K}_s$  are given here for both cases :

$$\mathbf{K}_{s,\text{left}} = \begin{pmatrix} \frac{EA}{4} & 0 & -\frac{EA}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{4} & 0 & \frac{EA}{4} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{EA}{4} & 0 & -\frac{EA}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{EA}{4} & 0 & \frac{EA}{4} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.14)$$

$$\mathbf{K}_{s,\text{right}} = \begin{pmatrix} \frac{EA}{8} & 0 & -\frac{EA}{8} & 0 & -\frac{EA}{8} & 0 & \frac{EA}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{8} & 0 & \frac{EA}{8} & 0 & \frac{EA}{8} & 0 & -\frac{EA}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{EA}{8} & 0 & \frac{EA}{8} & 0 & \frac{EA}{8} & 0 & -\frac{EA}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{EA}{8} & 0 & -\frac{EA}{8} & 0 & -\frac{EA}{8} & 0 & \frac{EA}{8} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (1.15)$$

We can see that the two cases are *not* equivalent. The matrix  $\mathbf{K}_{s,\text{left}}$  does not have the off-diagonal components present in  $\mathbf{K}_{s,\text{right}}$ . Those off diagonal components mean that

the reinforcement creates an interaction between nodes 2 and 3, as well as 1 and 4. This is logical, since in the left case, both reinforcements are independent : a force on node 2 would not cause a displacement of node 3 (only in terms of reinforcement stiffness ; the concrete stiffness matrix has an interaction between the two nodes). However, both cases react in the same way to an axial displacement, such as  $u_a = (0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0)^T$  :

$$\mathbf{K}_{s,\text{right}} u_a = \mathbf{K}_{s,\text{left}} u_a = \begin{pmatrix} -\frac{EA}{4} \\ 0 \\ \frac{EA}{4} \\ 0 \\ \frac{EA}{4} \\ 0 \\ -\frac{EA}{4} \\ 0 \end{pmatrix} \quad (1.16)$$

It can also be noticed that the equations of the discrete model are recovered from equation (1.10) as  $y$  goes to 1 or -1, which is consistent with the statement that the embedded model is a generalization of the discrete model.

## 1.2 Reinforcement behavior

### 1.2.1 Passive reinforcements

Generally, reinforcement in concrete is made of steel. In Switzerland nowadays, most reinforcement are fabricated with a 5% quantile yield limit of 500 N/mm<sup>2</sup>. There are three types of ductility that can be fabricated : A, B and C. A is the most fragile type and C the most ductile, in terms of strain before reaching the ultimate strength of the reinforcement.

Steel is a strain-hardening plastic material. The usual stress-strain curve looks like :

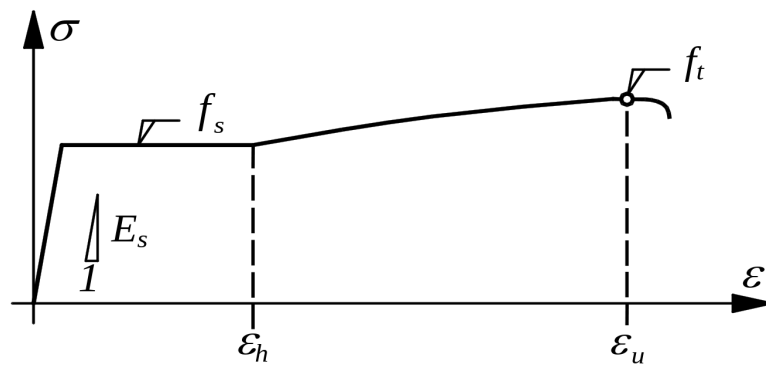


Figure 1.3: Typical stress-strain curve for reinforcement steel. Figure taken from [16].

The value of  $E_s$  used in the SIA 262 code is 205 GPa. For simplicity, reinforcement are often considered as an elastic, perfectly plastic material, meaning that the yield plateau occurring after the elastic phase is infinite. This is a good assumption in a lot of cases and allows for easy and efficient design. It may not be sufficient in the case of earthquake resistant structures, where strain hardening has an effect on the size of the plastic zones that dissipate the earthquake energy [14].

In the first part of this project, we will only consider a linear elastic constitutive law for the reinforcements. In a second part, we will study the behavior of the embedded model with non linear constitutive laws. For this, we use a linear isotropic hardening constitutive law. A numerical algorithm from [10] is described in appendix B.

### 1.2.2 Prestressing cables

The steel used for prestressing cables has a much higher resistance than the steel used for passive reinforcements. However, it does not have a yielding plateau, so the yield limit is defined as the stress needed to impose an irreversible strain of 0.1%. A typical stress-strain curve is shown below :

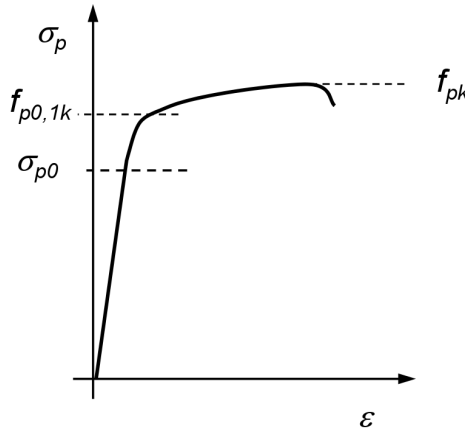


Figure 1.4: Typical stress-strain curve for prestressing cable steel. Figure taken from [15].

On this figure,  $\sigma_{p,0}$  is the stress applied to the cable, and it typically is equal to  $0.7f_{pk}$ . This is so that the cable reaches its yield limit  $f_{p0.1k}$  at about the same time as the passive reinforcement reaches its yield limit  $f_{sk}$ . Otherwise, the displacements required to activate the cable fully would be beyond the displacement capacity of the structure.

However, the stress in the cable is not constant over its length. There is some loss of stress due to friction in the sheath. This loss is given by the SIA 262 code.

$$\sigma_p(x) = \sigma_{p,0}e^{(-kx-\beta\theta)} \quad (1.17)$$

Where  $x$  is the position of a point on the cable,  $\theta$  is the cumulated change of angle of the cable,  $k$  and  $\beta$  are parameters of the sheath and the cable given by the manufacturer.

Most of the work presented in this thesis is done with the assumption that the steel reinforcements follow a linear elastic constitutive law. When studying non-linearity, an elastic-perfectly plastic constitutive law will be used.

## 1.3 Concrete behavior

Concrete is a cementitious material. It is highly non-linear, as it shows plasticity with hardening and softening. Usually, concrete is considered to have no resistance in tension,

even though it is resistant to a small extent. A typical stress-strain curve for concrete is shown below.

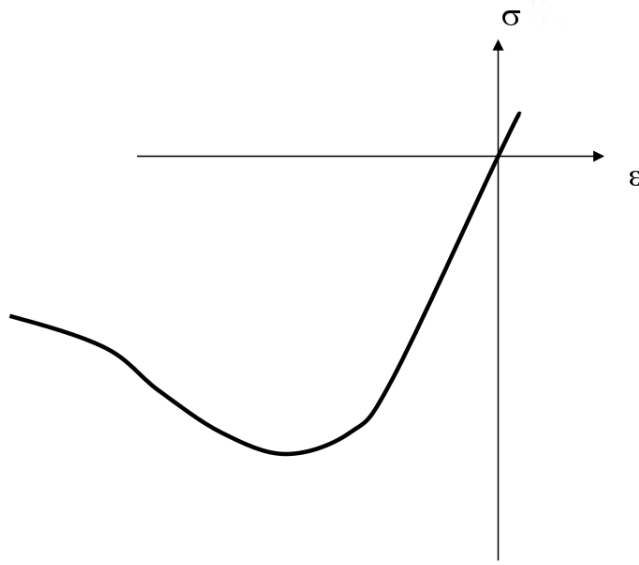


Figure 1.5: Typical concrete stress-strain curve. Figure taken from [16].

Some constitutive laws exist for concrete behavior. Gomes and Awruch use a modified version of Chen’s model in [13]. The equations of this model will not be covered in this thesis, as they go over the scope of this project. The code developed in this project will be compared to another code, *jconc* developed by the ibeton lab at EPFL. This code uses a 2D constitutive law from the modified compression field theory, by Vecchio and Collins [18].

For the majority of this project, the behavior of concrete will be taken as linear elastic. A non linear case is covered at the end of the report, which makes use of a regularized sequentially linear saw-tooth softening model from [17], described in appendix B.

# Chapter 2

## Embedded model, method and algorithm

The embedded reinforcement model requires more preprocessing than the other two models. Since the integral of equation (1.10) needs to be computed along the reinforcement, it is necessary to find the intersections of the reinforcement with the elements of the mesh. Then, in order to compute the integral, the integrand  $\mathbf{B}^T \mathbf{C}^T \mathbf{D} \mathbf{C} \mathbf{B}$  should be evaluated at the quadrature points along the reinforcement, and not the quadrature points of the concrete element. Finally, each reinforcement stiffness matrix  $\mathbf{K}_{s,i}$  should be either added to the local stiffness matrices, which would be later assembled into the global stiffness matrix, or directly assembled into the global stiffness matrix.

The algorithm implemented for the project is as follows :

1. User input :
  - Material properties for reinforcements
  - Geometry of reinforcements
2. Compute the intersections of the reinforcements with the mesh
3. Discretize the reinforcements into elements
4. For each element of the reinforcement :
  - (a) Compute the quadrature points
  - (b) Compute the constitutive matrix  $\mathbf{D}_s$
  - (c) Compute the cosine matrix  $\mathbf{C}$
  - (d) Compute the shape derivatives  $\mathbf{B}$
  - (e) Integrate equation (1.10)
5. Assemble the stiffness matrices of each element of the reinforcements

This algorithm was implemented into Akantu [1], a finite element library developed by the LSMS lab at EPFL. For the sake of simplicity, it was only implemented for 2D triangle linear elements and 3D tetrahedron linear elements. Another library was used for the computation of the intersections : the Computational Geometry Algorithm Library [3]. As

for the geometry of the reinforcements, even though the embedded model allows arbitrary shapes, only 2-nodes segment elements were considered during the project. Complex geometries can however be created by external software (e.g. a mesher), then discretized into segments and given to the algorithm for preprocessing. This allows for a certain ease in the definition of the reinforcement geometry, especially for parabolic cables.

## 2.1 Computing intersections

In order to efficiently compute the intersections of a reinforcement with the mesh of concrete elements, we use CGAL's Axis-Aligned Bounding Box algorithm [7]. This algorithm is a divide-and-conquer algorithm that helps to compute the intersection of one query object (in this case a segment) with a large number of geometrical primitives (the elements of the mesh). The mesh is here decomposed into triangles, whether it is a 2D mesh or a 3D mesh, due to CGAL limitations, then the AABB-tree algorithm forms a tree of boxes that encompass smaller parts of the mesh at deeper levels of the tree, as shown in the following figure.

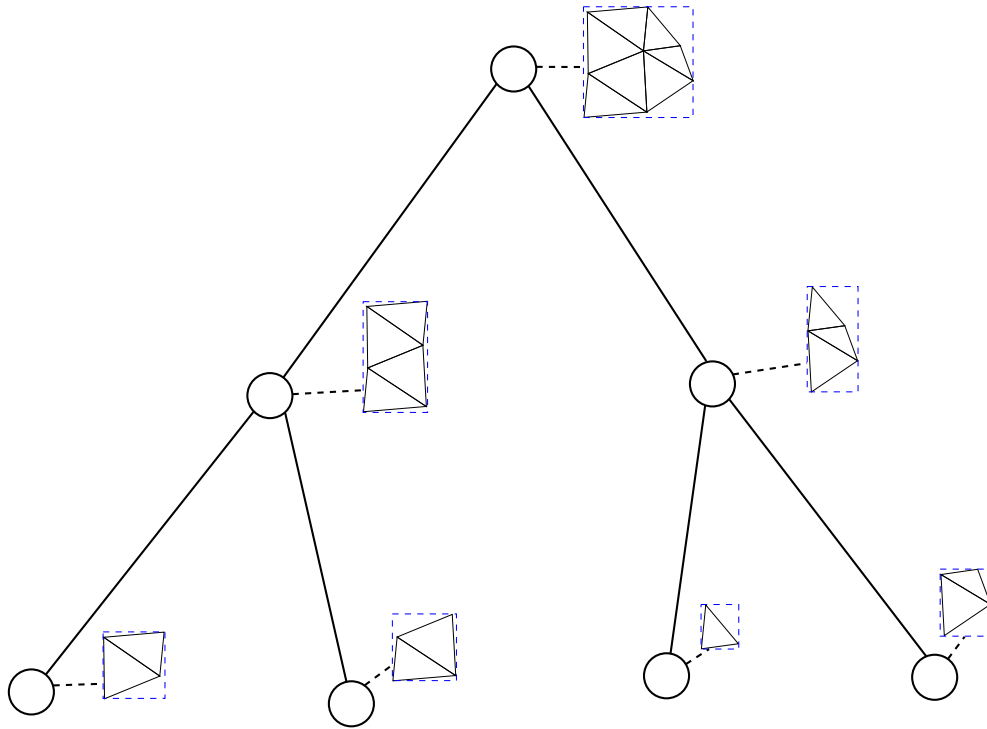


Figure 2.1: Example of AABB-tree structure (only three levels). Bounding boxes are shown in dashed blue lines.

Determining whether a bounding box is intersected by a segment is computationally cheap. It also allows the intersection algorithm to completely ignore parts of the mesh, leading to a good algorithmic performance, with a complexity of  $O(\log(n))$ ,  $n$  being the total number of primitives.

Because of CGAL limitations, it is impossible to directly compute the intersection of a segment and a tetrahedron, which is needed in 3D meshes. In order to overcome this problem, we decompose the tetrahedra into triangles. The tree is constructed with triangles as its base primitive. The AABB-tree algorithm returns, for the intersection with a segment, a list of all the intersection points, along with a list of element ids, representing the tetrahedra that were intersected. With this, the segments that are inside the tetrahedra can be reconstructed.

## 2.2 Stiffness matrix

The next step in the algorithm is to compute the quadrature points for the reinforcement. By creating a `Mesh` object in Akantu for the reinforcement, the `FEEngine` class can compute the quadrature points. In the case of straight linear reinforcement elements, the quadrature point is the middle of the segment.

It is then necessary to compute  $\mathbf{D}_s$ ,  $\mathbf{C}$  and  $\mathbf{B}$  on the quadrature points. Since the reinforcements are straight and the constitutive law is linear elastic, the computations of  $\mathbf{D}_s$  and  $\mathbf{C}$  are trivial. For higher order shapes of reinforcements, the knowledge of the parametrization  $\vec{r}$  is necessary to compute the directing cosines  $l$ ,  $m$  and  $n$ . The computation of the shape derivatives  $\mathbf{B}$  is also straightforward, although it was not possible to evaluate the shape derivatives on arbitrary points in Akantu, so this feature was added to the library.

Once the product  $\mathbf{B}^T \mathbf{C}^T \mathbf{D}_s \mathbf{C} \mathbf{B}$  has been computed on the quadrature points, the `FEEngine` class can integrate equation (1.10) over the reinforcement. At the end of the algorithm, the program has an array of local stiffness matrices (one for each reinforcement element), and assembles that array into the global stiffness matrix. Then the system  $\mathbf{K}u = f$  is solved for the displacements at the nodes of the mesh (with the appropriate boundary conditions).

## 2.3 Residual forces

It is also necessary to be able to compute stresses in the reinforcement elements, as well as the residuals of the system, in order to have the necessary data for the design of a structure. Residuals are also used in the case of non-linear materials, because they are a central part of the Newton-Raphson method for convergence.

Since a perfect bound is assumed between the steel and the concrete, the axial strain of the reinforcement is given by equation (1.5). The stress in the reinforcement is then computed with the constitutive equation :

$$\sigma_s = \mathbf{D}_s \varepsilon_s \quad (2.1)$$

Where  $\mathbf{D}_s$  represents the constitutive law (and can include non-linear effects). The integration of the residuals is done using the following equation :

$$\vec{r}_{s,i}^e = A_{s,i} \int_S \mathbf{B}^T \mathbf{C}_i^T \vec{\sigma}_s \, ds \quad (2.2)$$



Where  $\vec{\sigma}_s$  is the reinforcement stress tensor in Voigt notation, and  $\vec{r}_{s,i}^e$  is the residual vector (defined on the nodes of the element  $e$ ), which needs to be assembled in the global residual vector. This vector should be zero on every degree of freedom, except the ones that blocked by a Dirichlet boundary condition. Then, the residual forces are the support reactions. This will be useful when comparing simulation results with analytical solutions of stresses.

# Chapter 3

## Comparison with discrete model

In this chapter, the method implemented will be applied to a simple example, and its results will be compared with those given by the program *jconc* developed by the ibeton lab at EPFL [4]. This program implements the discrete reinforcement model and the modified compression-field theory from [18]. However, since the method implemented for this project is only linear elastic, material properties will be tweaked in *jconc* in order to stay in the elastic regime.

### 3.1 Comparison example

The considered example is a square patch of concrete of dimension 1 m by 1 m. It has only one horizontal reinforcement at mid-height. Horizontal displacement is blocked on the bottom side, and vertical displacement is blocked on the left side. The right side is subject to a pressure of 1000 Pa, while the top side is free. The example is shown in figure 3.1.

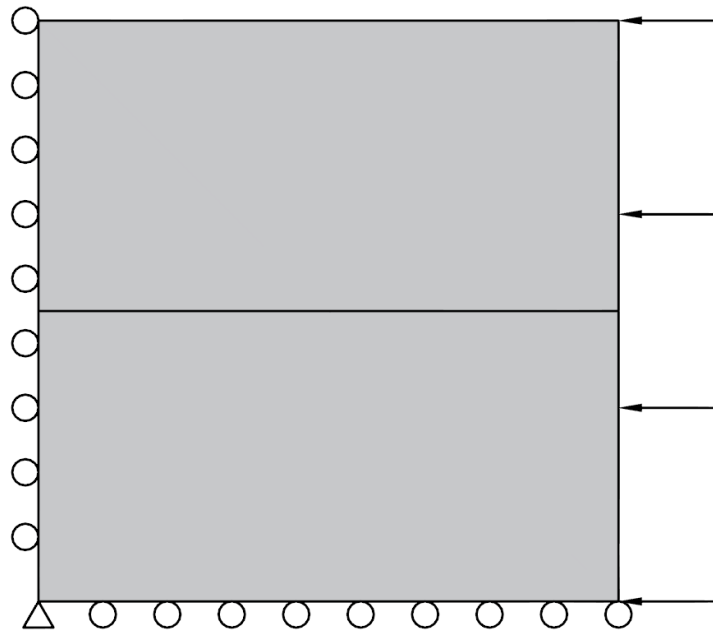


Figure 3.1: Example used for comparison

The materials used have the following properties :

Table 3.1: Material properties.

	Concrete	Reinforcement
E	30 GPa	205 GPa
$\nu$	0	-
A	-	0.1 m <sup>2</sup>

The mesh used for the comparison is shown in figure 3.2. It is a very coarse mesh, but allows for easy matching of the data between the code developed with Akantu and *jconc*.

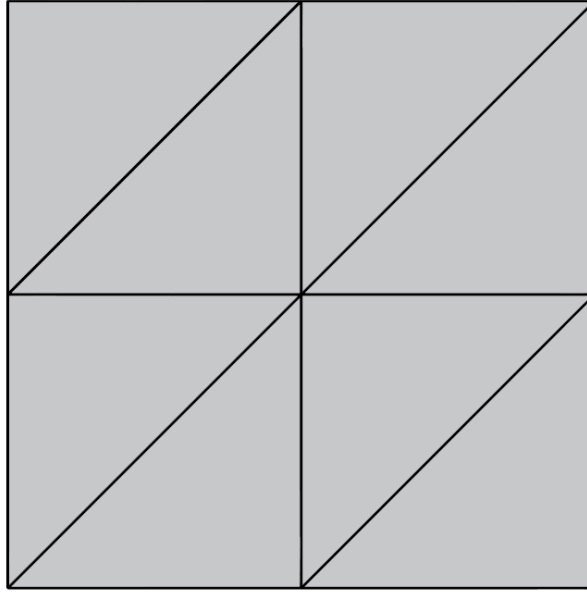


Figure 3.2: Mesh used for the example

With that mesh, the reinforcement goes through the nodes of the mesh. The discrete and embedded models should therefore give the same results. Figures 3.3 and 3.4 show the deformed shape of the example.

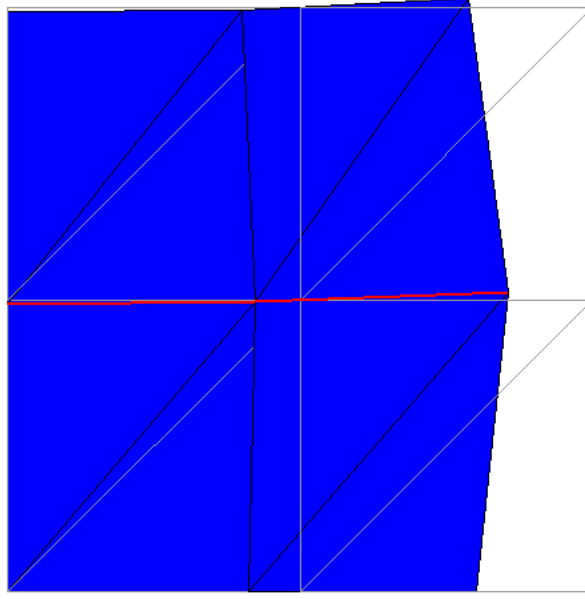


Figure 3.3: Deformed shape using jconc. Scale factor of  $8 \cdot 10^6$ .

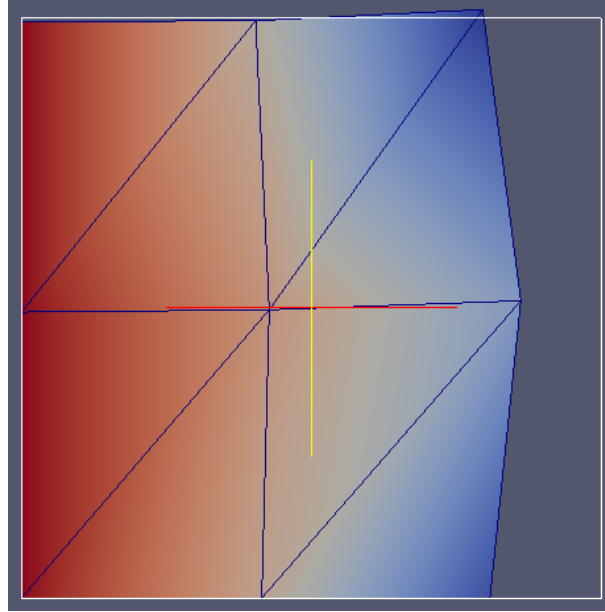


Figure 3.4: Deformed shape using Akantu. Scale factor of  $8 \cdot 10^6$ . Colors shows the horizontal displacement.

From those figures, the deformed shape is very similar. By comparing the displacements at the nodes, it is confirmed that the two codes are strictly equivalent for this particular mesh, with the nodes aligned with the reinforcement.

Table 3.2: Node displacements

Position	<b>jconc</b>		<b>Akantu</b>	
	$u_x[\cdot 10^{-8} \text{ m}]$	$u_y[\cdot 10^{-9} \text{ m}]$	$u_x[\cdot 10^{-8} \text{ m}]$	$u_y[\cdot 10^{-9} \text{ m}]$
(0, 0)	0	0	0	0
(0.5, 0)	-1.075	0	-1.075	0
(1, 0)	-2.408	0	-2.408	0
(0, 0.5)	0	-0.8637	0	-0.8637
(0.5, 0.5)	-0.9136	-0.3656	-0.9136	-0.3656
(1, 0.5)	-1.742	1.5945	-1.742	1.5945
(0, 1)	0	-0.7589	0	-0.7589
(0.5, 1)	-1.207	-0.5494	-1.207	-0.5494
(1, 1)	-2.562	1.858	-2.562	1.858

This is caused by the equality of the global stiffness matrices, as predicted.

## 3.2 Convergence analysis

In order to find out how effective the embedded model is at representing the contribution of the reinforcement, it is necessary to establish if the method converges, and compare the rate of convergence of the method with that of the discrete model.

To study the convergence of the embedded model, the example shown in figure 3.1 is used with different unstructured meshes, with no guaranteed node aligned with the reinforcement. Figure 3.5 shows one of the meshes and the elements that are affected by the reinforcement.

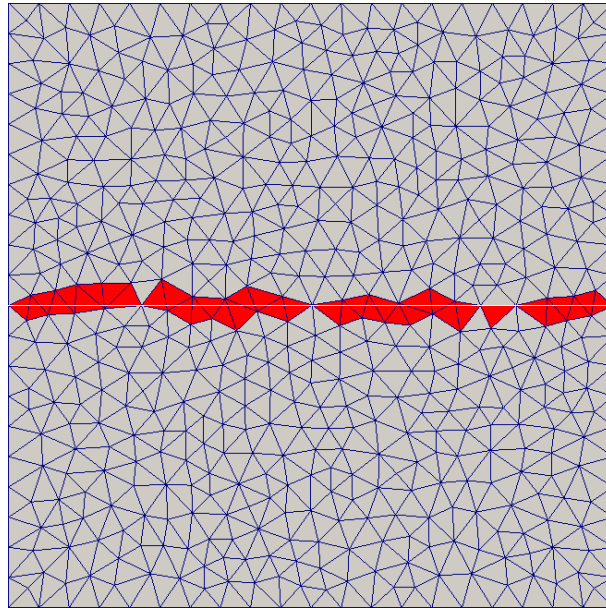


Figure 3.5: Unstructured mesh example. Red elements are affected by the reinforcement.

This figure shows that when nodes are not aligned with the reinforcement, the effect it has on the stiffness matrix is dependent on the mesh, and the elements affected do not

form a straight line. This behavior is however expected to improve as the mesh gets finer.

In order to assess the convergence of the method, we look at the horizontal displacement on the right edge of the plate, where the load is applied, i.e.  $u_x(1, y)$  as  $y$  varies. Figure 3.6 shows this function with different meshes.

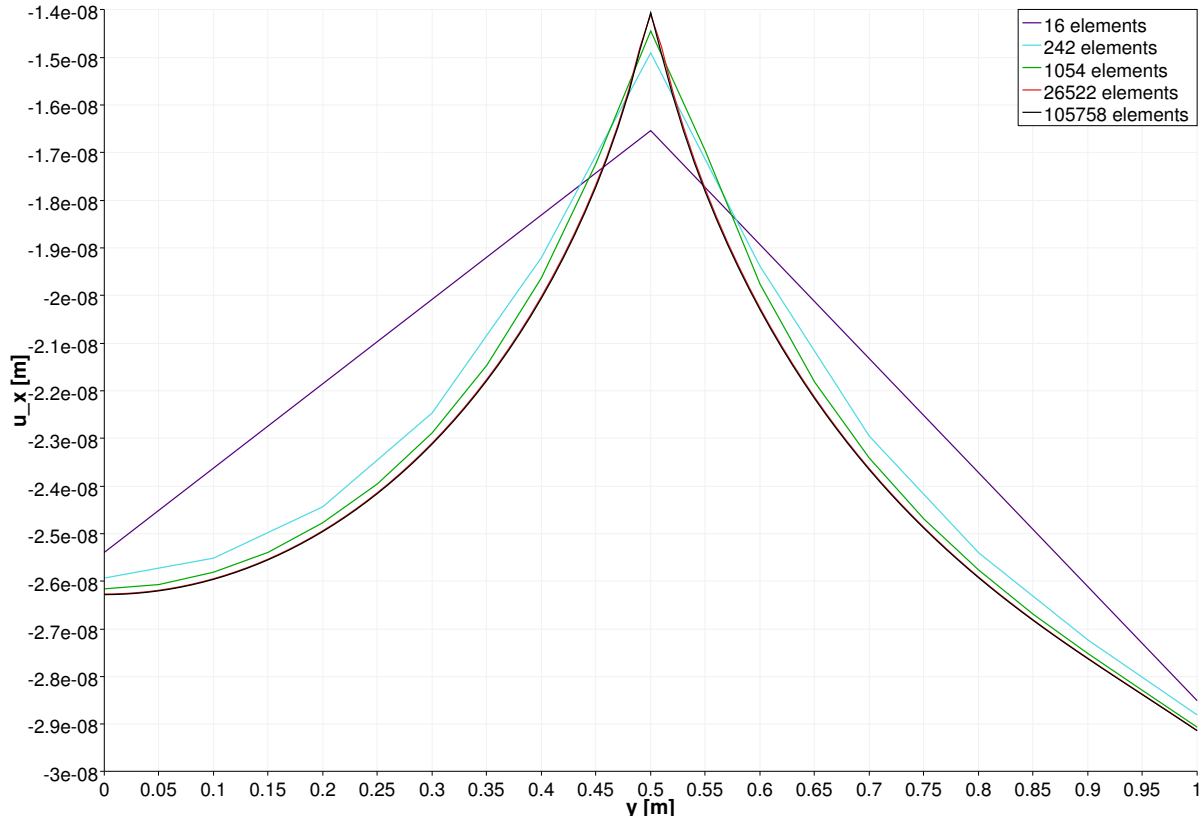


Figure 3.6: Horizontal displacement of the right edge of the plate.

The effect of the reinforcement is clearly seen on this figure, as the absolute value of the displacement is lower at  $y = 0.5$  m, due to the increased stiffness of the reinforcement. This figure shows that the method converges as the size of the mesh gets smaller : the colored curves tend to the black curve.

In order to analyze the rate of convergence of the embedded model and compare it with the rate of convergence of the discrete model, it is necessary to have a reference to compute an error. Since finding an analytical solution to the example of figure 3.1 is tedious, it was decided to take a reference computed with the discrete model and a very fine mesh (660640 elements).

The comparison is done by decreasing the size of the mesh for both models and computing an error with the reference case, in terms of potential energy.

$$e(\tilde{u}) = \frac{|E_p(\tilde{u}) - E_p(u_{\text{ref}})|}{E_p(u_{\text{ref}})} \quad (3.1)$$

For practical reasons, the discrete model computations will be done with the algorithm developed, but using a modified mesh so that the reinforcement goes through the nodes of

the mesh, thus taking advantage of the fact that the embedded model is a generalization of the discrete model. Figure 3.7 shows the error for both models as the size of the mesh decreases.

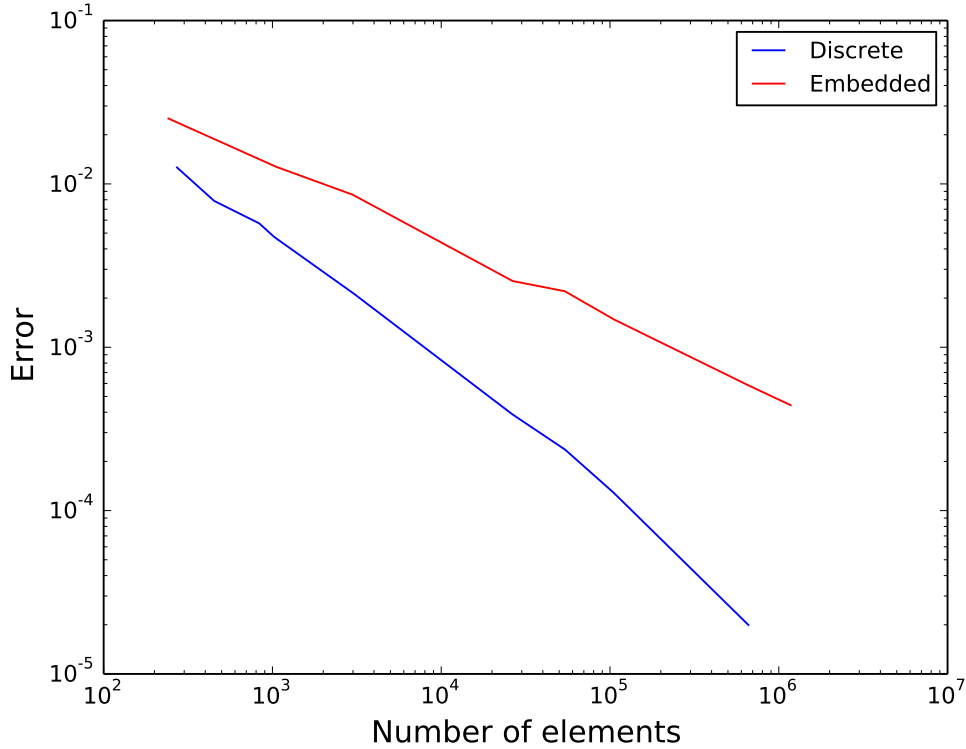


Figure 3.7: Potential energy relative error as a function of the number of elements, for the discrete and embedded models.

On this plot, it is clear that the embedded model does not converge as fast as the discrete model. This is because the effect of the reinforcement on the elements, computed using the embedded model, never forms a straight line, as shown in figure 3.5.

It seems however that the two methods are converging to the same point, as the potential energy plot in figure 3.8 shows.

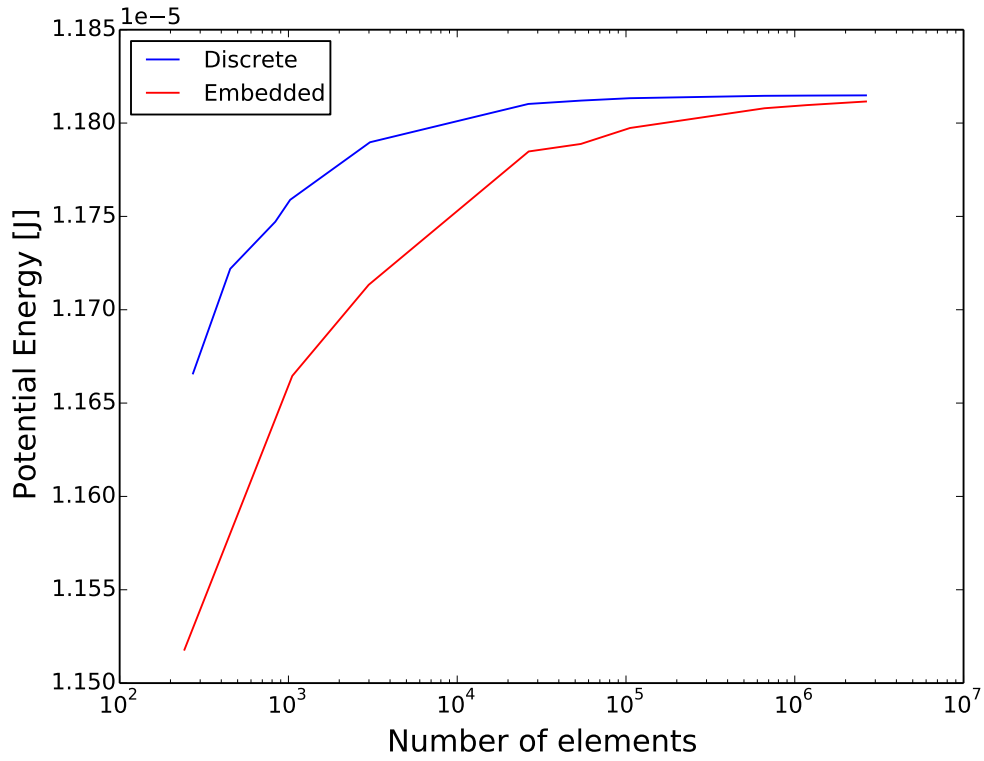


Figure 3.8: Potential energy as a function of the number of elements, for the discrete and embedded models.

Figures 3.7 and 3.8 are essential in validating and assessing the efficiency of the embedded model. Figure 3.8 shows that the embedded model tends to the same solution as the discrete model, thus validating the use of the embedded model for design. However, the rate of convergence is much lower for the embedded model : reaching the same level of approximation as the discrete takes more than 10 times the number of elements.

In order to find if the convergence rate is correlated to the mesh, the same analysis of the embedded and discrete model was conducted with a structured (periodic) mesh for the embedded model. The results are given in figures 3.10 and 3.9.



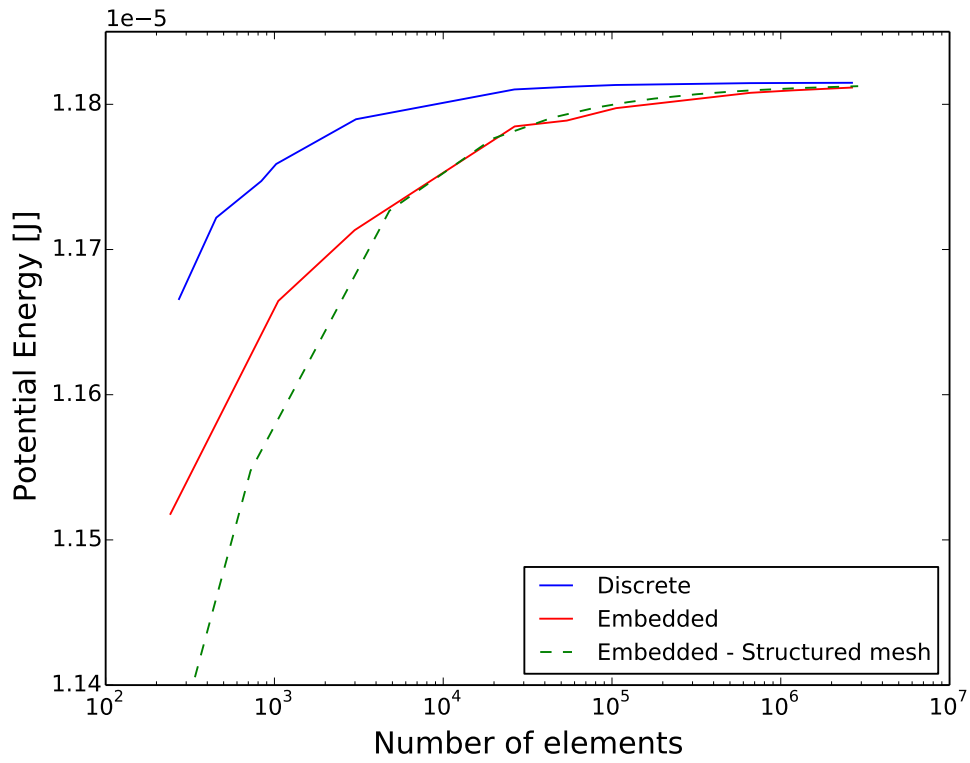


Figure 3.9: Potential energy comparison between the three cases.

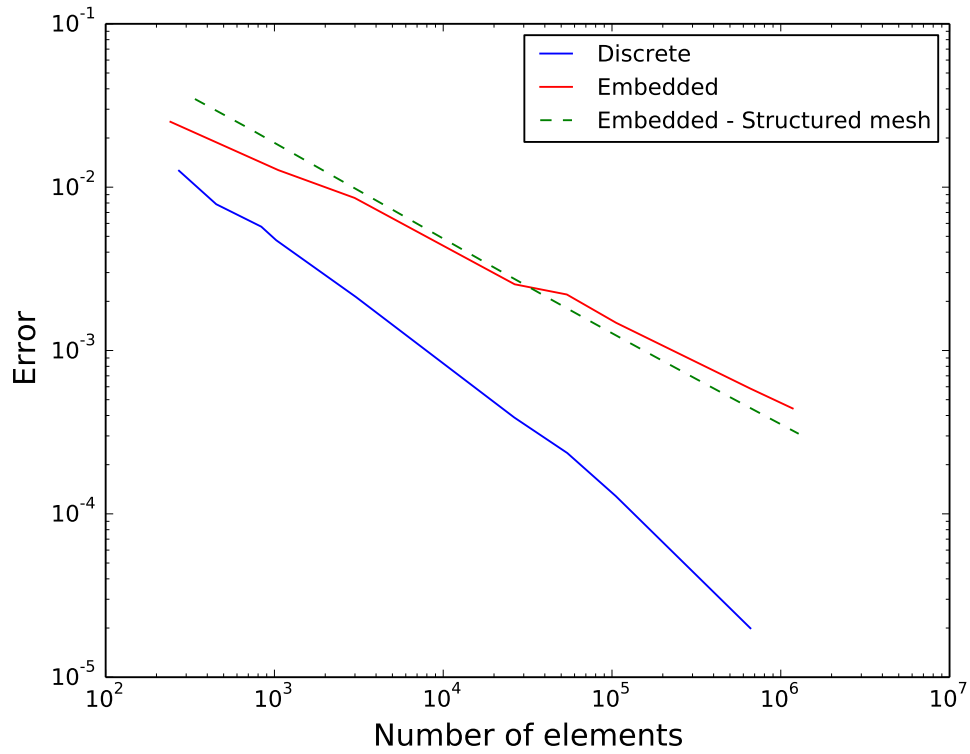


Figure 3.10: Convergence comparison between the two cases.

The convergence rate of the embedded model with a structured mesh is better than when a unstructured mesh is used. The error curve for the structured mesh is also straighter than the curve for a unstructured mesh. This is due to the random process behind the generation of unstructured meshes. This randomness can affect the error to a small extend. The rate of convergence of the embedded model is therefore affected by the randomness of the mesh. On a smaller scale, it can be said that the quality of the results depends on the way the reinforcement intersects the concrete elements. This process is further explained in chapter 4.

# Chapter 4

## Prestressing

Be it through prefabricated, prestressed concrete elements or large concrete structures post-tensioned with cables, pre/post-stressing is a major technique in concrete construction. It allows a good economy of material (steel or concrete), less cracks, which, with the right design, can lead to a higher lifetime, and less deformations. With the very large number of prestressed existing structures that may need monitoring, it was important to include prestressing in the models developed during this project.

### 4.1 Effect of prestressing

Prestressing can be taken into account in hand calculations in two different ways :

- As external forces
- As an eigenstress

Considering that the prestressing is an external load means that the pre/post-tension elements (cables or reinforcements) are not considered in a tension state. The effect of prestressing is translated into equivalent external forces.

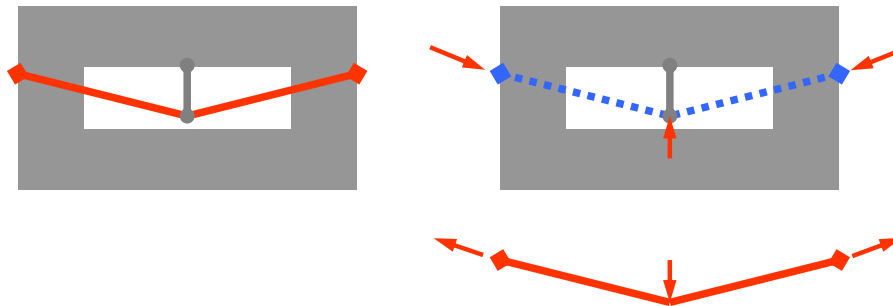


Figure 4.1: Effect of post-tension cable as eigenstress (left) and external forces (right). Figure taken from [15].

The external forces applied are of two kinds : the anchor forces at the ends of the cable, and the deviation forces at each change of angle of the cable, that are a result of equilibrium. Since prestressing is often used to balance the self weight of a structure and

the permanent loads (which are often constant distributed loads), the cable is given the shape of a parabola, so that the deviation forces are a constant distributed load, oriented upwards.

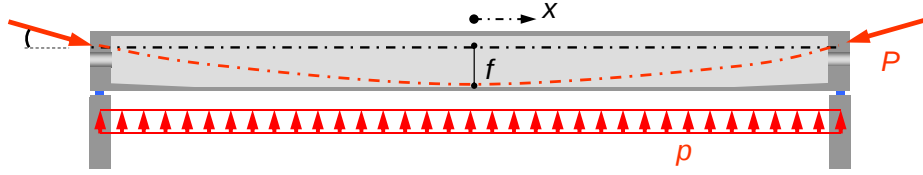


Figure 4.2: Replacement forces for parabolic post-tension cable. Figure taken from [15].

In this project, it was assumed that, in the case of post-tension cables, there is no friction between the cable and its sheath. In reality, the friction causes a loss of stress along the cable. The SIA 262 code gives the following formula :

$$\sigma_p(x) = \sigma_p e^{-kx - \mu\beta} \quad (4.1)$$

Where  $\sigma_p$  is the initial stress,  $kx$  represents the loss of stress due to the length of the cable, and  $\mu\beta$  is the loss due to the change of angle of the cable (it would be zero for a straight cable). Due to lack of time, this formula was not implemented in the model. However, since the stress created by the tension in the cable is stored on every quadrature point, it would only require a layer of preprocessing to compute the correct stress on the quadrature points with equation (4.1).

The second way to consider prestressing in a structure is the one selected for the finite element model. The prestress is considered as stress to be added to the stress of the reinforcement given in equation (2.1).

$$\sigma_s = \mathbf{D}_s \varepsilon_s + \sigma_p \quad (4.2)$$

Like previously mentioned, this equation is evaluated at every quadrature point, for the integration of the residuals (2.2). The residuals are then used as the right-hand member of the linear equation system  $\mathbf{K}u = \mathbf{f} - \mathbf{r}$ .

## 4.2 Numerical application

In order to verify the validity of this model, we compare the results to an analytical solution. Since we could not find analytical solutions in terms of displacements, we will compare the stresses given by the finite elements model to a stress distribution derived from analytical solutions. The considered problem is the following :

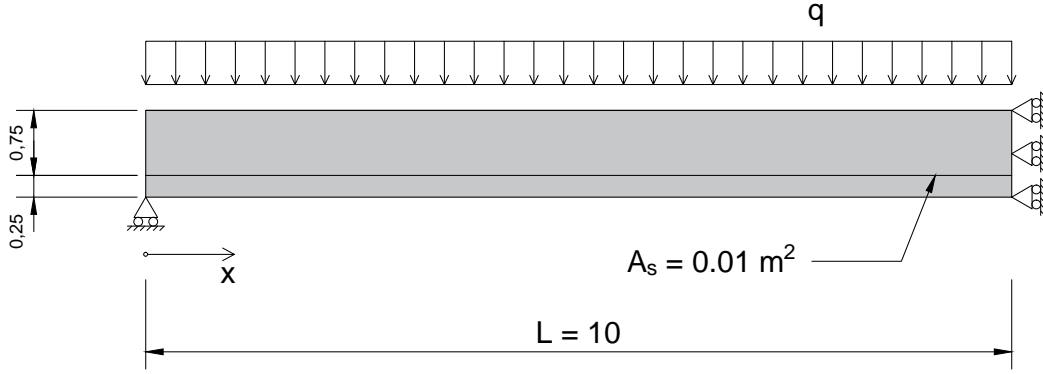


Figure 4.3: Beam example. Dimensions are in meters.

Dimensions are in meters. In this section, we will study the convergence of different cases :

- Beam is uniformly loaded ( $q = 1 \text{ kN/m}$ ) :
  - With no reinforcements
  - With reinforcements (discrete model)
  - With reinforcements (embedded model)
- Beam is not loaded, but reinforcement is prestressed :
  - Discrete model
  - Embedded model

We use those two cases to assess the ability of the embedded model to predict stresses for a beam with passive and active reinforcements. The materials used have the following properties :

Table 4.1: Material properties.		
	Concrete	Reinforcement
E	30 GPa	210 GPa
$\nu$	0	-
A	1 m <sup>2</sup>	0.01 m <sup>2</sup>
$\sigma_p$	-	1 MPa

### 4.2.1 Analytical solutions

Derivation of the analytical solution is done using the equivalent area method. We suppose that the beam is subject to a compressive force  $P$ , applied at the neutral axis, and a moment  $M$ . Using the equivalent area, we can derive the analytical stress distribution. The equivalence coefficient is :

$$n = \frac{E_s}{E_c} = 7 \quad (4.3)$$

From  $n$  are derived the equivalent area, static moment and moment of inertia.

$$\underline{A} = A_s + \frac{1}{n}A_c \quad (4.4)$$

$$\underline{S} = S_s + \frac{1}{n}S_c \quad (4.5)$$

$$\underline{I} = I_s + \frac{1}{n}I_c \quad (4.6)$$

With the moments of inertia computed from the composite neutral axis :

$$y_G = \frac{\underline{S}}{\underline{A}} \quad (4.7)$$

We know that the initial strain in the cable is :

$$\varepsilon_{s,p} = \frac{\sigma_p}{E_s} \quad (4.8)$$

With  $\sigma_p$  the prestress in the reinforcement. The total strains in the reinforcement and the concrete are :

$$\varepsilon_s = \varepsilon_{s,p} + \bar{\varepsilon} + \varepsilon^* \quad (4.9)$$

$$\varepsilon_c = \bar{\varepsilon} + \varepsilon^* \quad (4.10)$$

Where  $\bar{\varepsilon}$  and  $\varepsilon^*$  are the strains due to  $P$  and  $M$  respectively.

$$\bar{\varepsilon} = \frac{-P}{E_s A_s + E_c A_c} \quad (4.11)$$

$$\varepsilon^*(y) = \frac{-M}{E_s I_s + E_c I_c} (y - y_G) \quad (4.12)$$

Then we can derive expressions for the stresses in the steel and concrete :

$$\sigma_s = \sigma_p - \frac{P}{\underline{A}} - \frac{M}{\underline{I}} (y_s - y_G) \quad (4.13)$$

$$\sigma_c(y) = -\frac{P}{n\underline{A}} - \frac{M}{n\underline{I}} (y - y_G) \quad (4.14)$$

Where  $y_s$  is the position of the steel cable. The stresses considered in the studies cases are shown in the following figures :

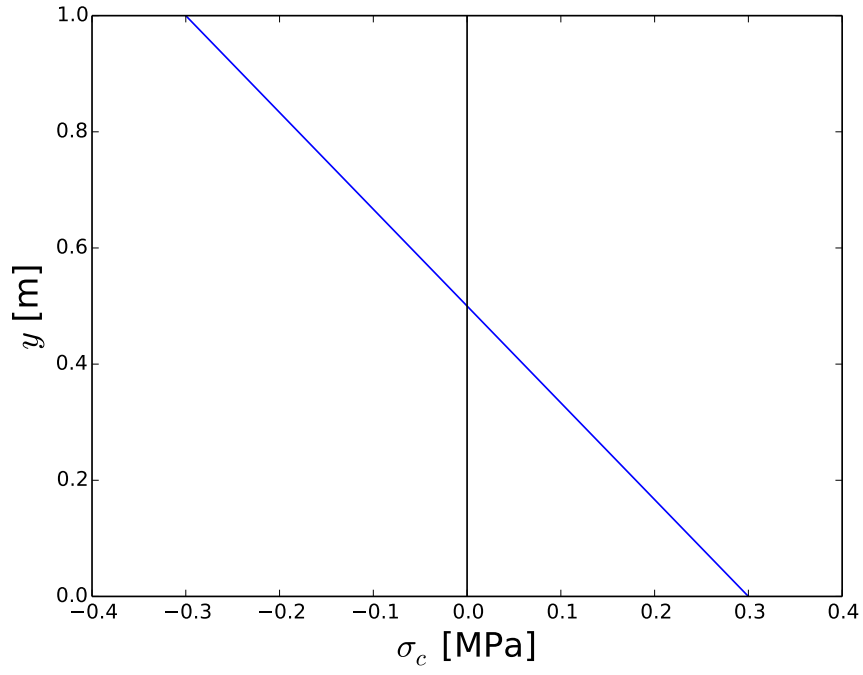


Figure 4.4: Analytical stresses for a load of  $q = 1$  kN/m with no reinforcements.  $P = 0$  kN and  $M = \frac{qL^2}{2} = 50$  kNm.

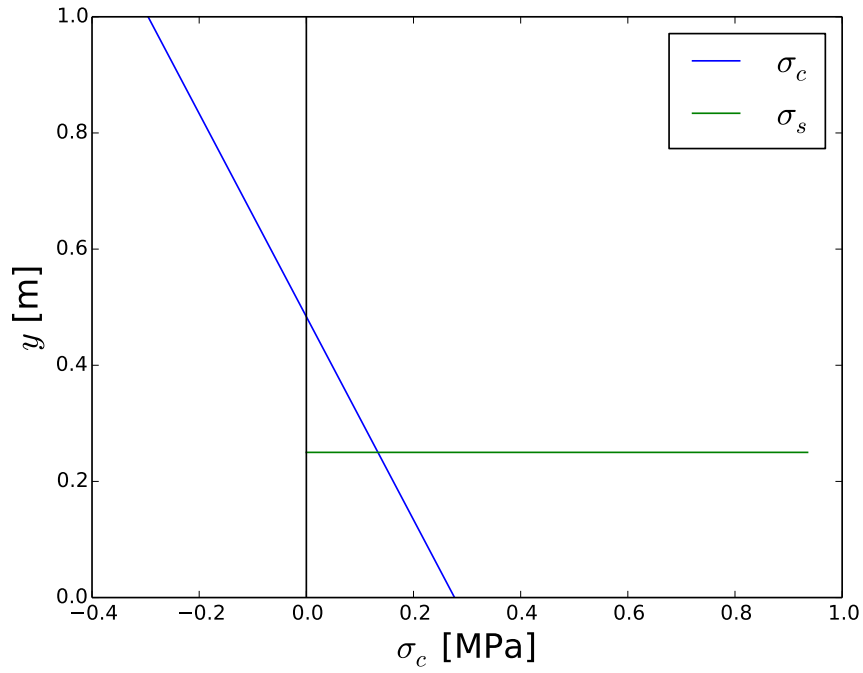


Figure 4.5: Analytical stresses for a load of  $q = 1$  kN/m with reinforcements.  $P = 0$  kN and  $M = \frac{qL^2}{2} = 50$  kNm.

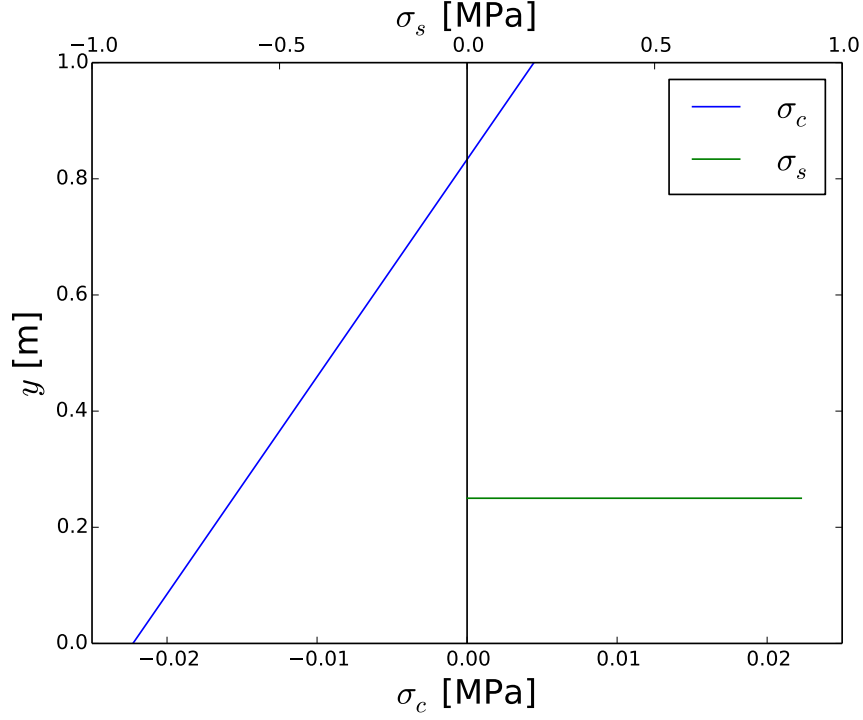


Figure 4.6: Analytical stresses for prestressed beam.  $P = \sigma_p A_s = 10$  kN applied at  $y_s = 0.25$  m.

### 4.2.2 Convergence analysis

In order to compute an error between the model and the analytical solution, we use the  $L_1$  norm of the stresses over the cross-section at  $x = 10$  m :

$$\|\sigma\|_1 = \int_A |\sigma_c| dA + |\sigma_s| A_s \quad (4.15)$$

The  $L_1$  norm of the stresses in the finite elements simulation is computed through the residuals on the nodes of the  $x = 10$  m edge :

$$\|r\|_1 = \sum_{i=1}^N |r_{c,i}| + |r_s| \quad (4.16)$$

The error is then computed as :

$$e = \left| \frac{\|r\|_1 - \|\sigma\|_1}{\|\sigma\|_1} \right| \quad (4.17)$$

The following figures show the error relative to the number of elements on the right edge of the problem.



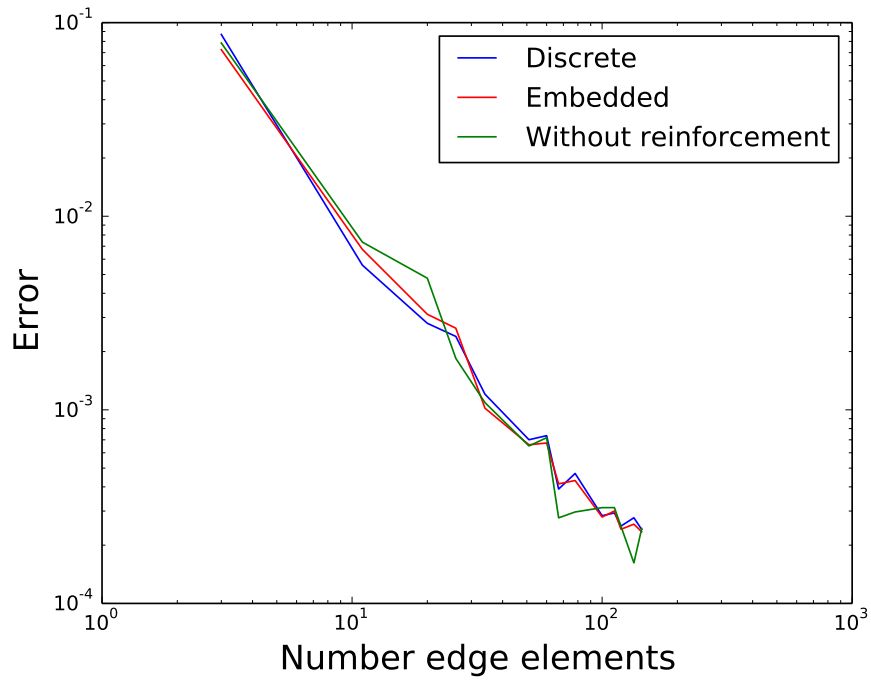


Figure 4.7: Convergence for beam with distributed load.

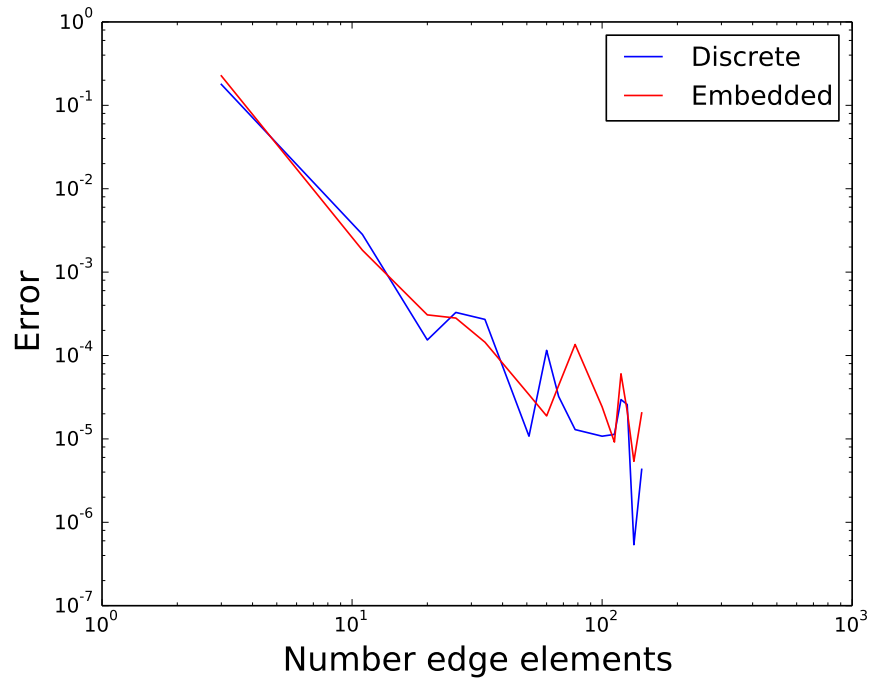


Figure 4.8: Convergence for prestressed beam with no external load.

These plots show an interesting phenomenon. Although the error seems to be decreasing as the number of elements decreases, it does not follow a straight line like the potential

energy error in figure 3.8. Rather, the error seems to oscillate. It seems that the chosen error criterion, based on the  $L_1$  norm of the stresses does not converge for this particular case. Figure 4.9 shows the potential energy for the prestressed beam. It seems that both models converge in terms of energy, but the residuals computed do not seem to converge, even though the error is low enough for practical purposes ( $10^{-5}$ ). This is emphasized by the fact that the error in a model with no reinforcements also oscillates. In the potential energy plot, the discrete model has a monotonous behavior, whereas the potential energy of the embedded model has not. This phenomenon is not understood at this time and would need further investigations.

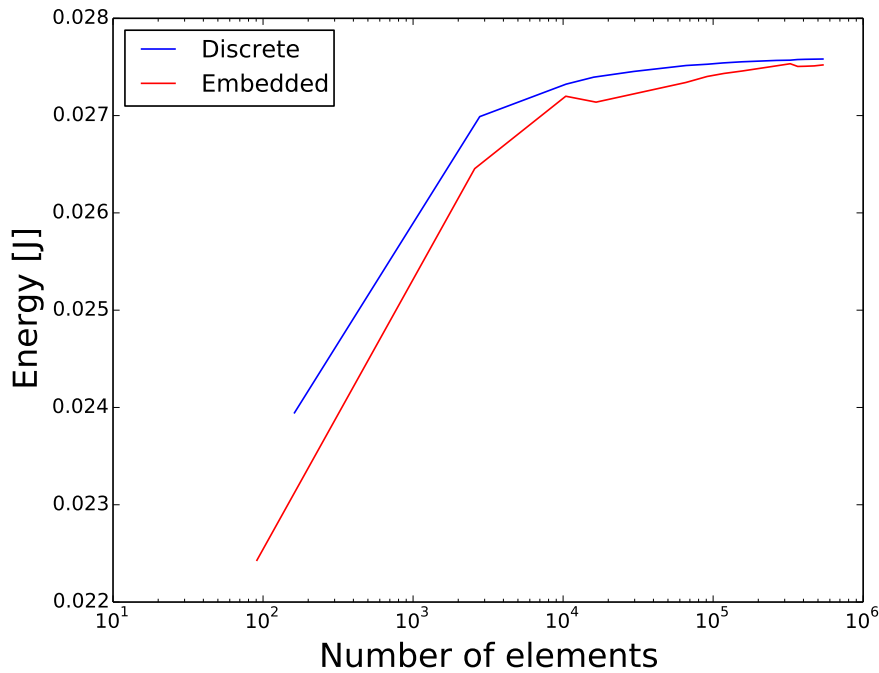


Figure 4.9: Potential energy of prestressed beam with no external load.

### 4.3 Stress distribution in reinforcement

This section will focus mainly on the stress distribution in the reinforcement and how it varies with the mesh size for the two models. The simulation studied is the same reinforced concrete beam with no load and a prestressing of 1 MPa in the reinforcement. Figures 4.10 to 4.13 show the stress along the reinforcement for coarse and fine meshes. The curves corresponding to fine meshes (figures 4.11 and 4.13) exhibit the same plateau of stress. This plateau is lower than  $\sigma_p = 1$  MPa because of the loss of prestress due to the compression and bending of concrete, see equation (4.13). However, the embedded stress distribution of figure 4.13 exhibits a noise phenomenon, whereas the discrete stress distribution is smooth.

This noise phenomenon becomes more important as the mesh gets coarser, as figure 4.12 shows. Here the stress plateau can be observed, but the oscillations of the curve around it are of bigger amplitude than in the discrete model. The behavior of the reinforcement is also not accurately represented for  $x < 1$  m. This noise can be explained

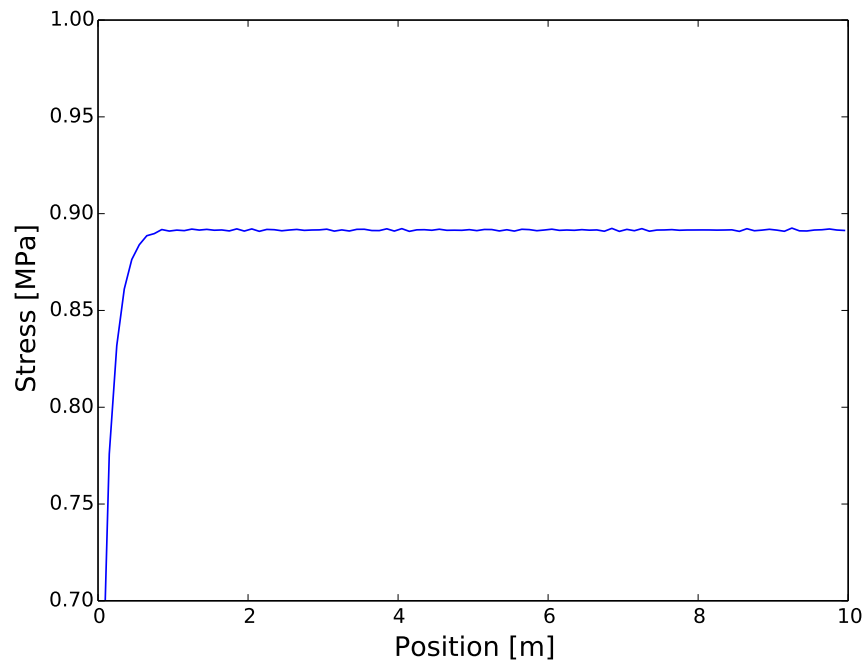


Figure 4.10: Stress along the reinforcement, discrete model. 5751 elements.

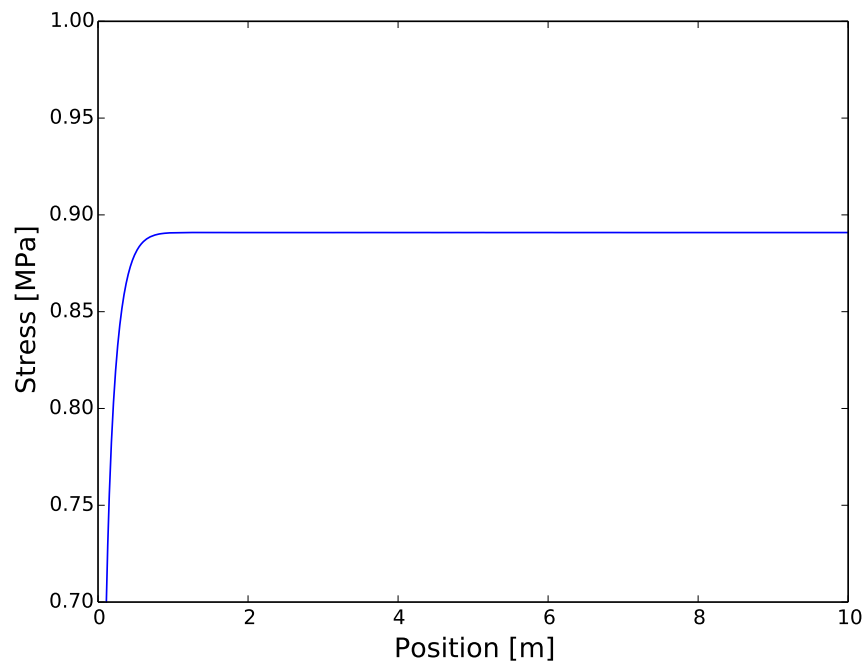


Figure 4.11: Stress along the reinforcement, discrete model. 1082813 elements.

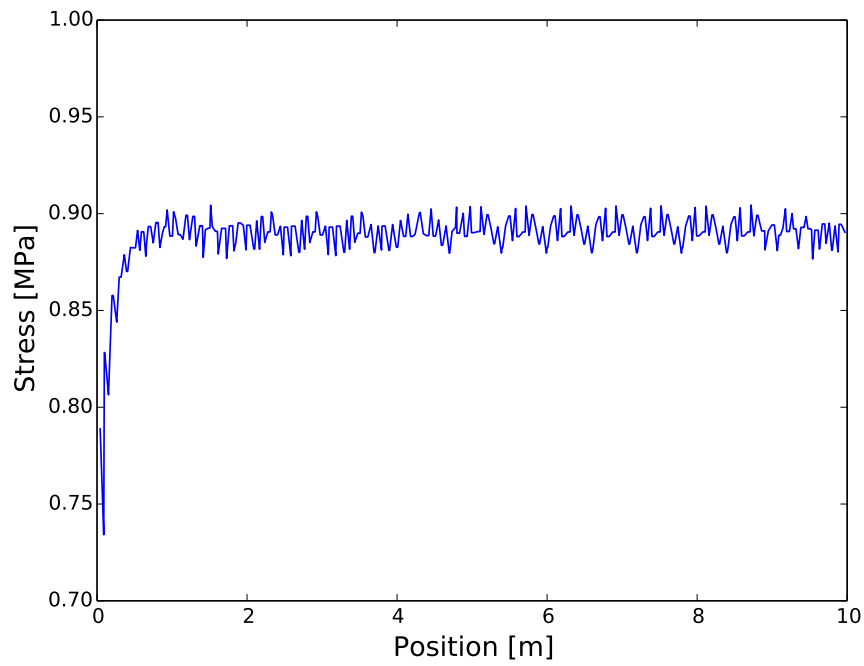


Figure 4.12: Stress along the reinforcement, embedded model. 5755 elements.

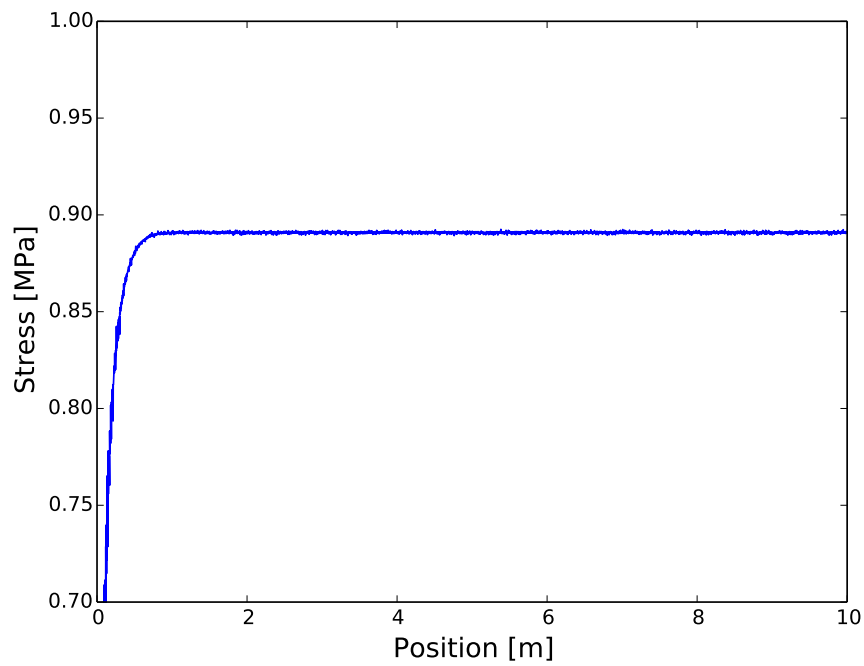


Figure 4.13: Stress along the reinforcement, embedded model. 1083121 elements.

by the way the reinforcement is intersecting with the concrete elements, as shown in figure 4.14.

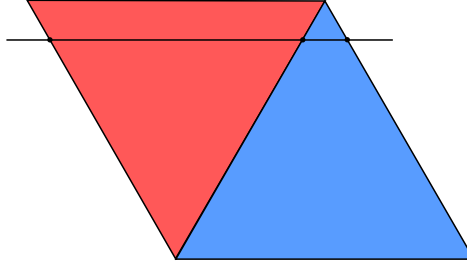


Figure 4.14: Reinforcement crossing elements with two different stresses.

In this figure, the reinforcement is crossing two elements whose center of gravity is not located at the same height. Therefore, according to equation (4.13), they have different strains. Since linear elements are used, the strains are constant in one element. Reinforcement stress being derived from the concrete element's strain, there is a jump in the former, causing this noise observed in the previous figures. This noise can potentially be greatly reduced with the use of second order elements for the concrete, as the strain in the element becomes linear.

Based on these observations, it can be said that the embedded model can be used to compute stresses in the reinforcements, as shown in 4.2.1, but the noise that appears forces the user to select fine meshes in order to have a good understanding of the reinforcement behavior, whereas the discrete model does not exhibit noise, even for relatively coarse meshes.

# Chapter 5

## Application to concrete bridge

This section aims to show the application of the embedded model to the real-life example of a concrete bridge. We first describe the bridge and its reinforcement layout, including post-tensioning cables, then compute the fatigue stress variation  $\Delta\sigma$  when a traffic load passes on the bridge, which is used to verify the bridge lifespan in the code SIA 262.

### 5.1 Description of the bridge

#### 5.1.1 Geometrical and material characteristics

This bridge is taken from a bridge studied in Prof. Muttoni’s class “Ponts en béton” given at a master level in EPFL’s civil engineering department. The bridge is a concrete beam on three pin supports, with two spans of 56.20 m. The cross-section is shown in figure 5.1.

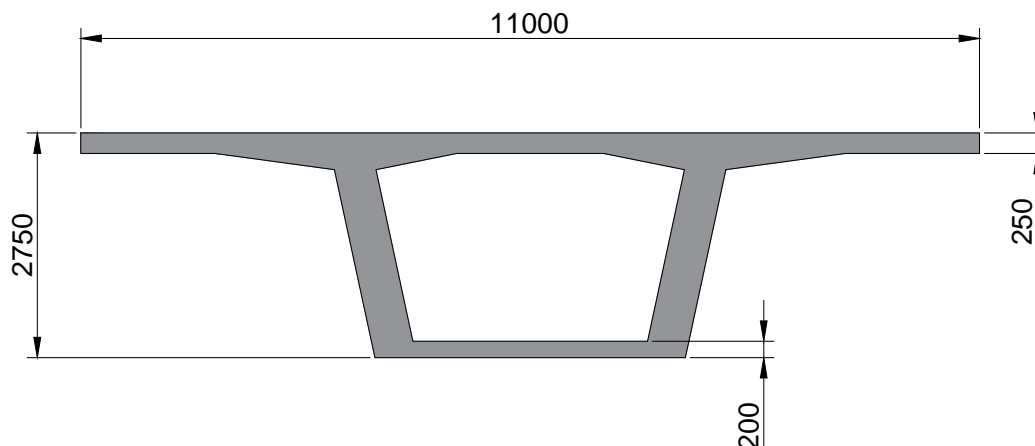


Figure 5.1: Bridge cross section. Dimensions in mm.

For simplicity reasons, this cross-section is used on the entire length of the bridge. In a real bridge, the lower part of the cross-section would be thicker on the support to carry the compressive load due to a negative moment. On the central support, the inside of the cross-section is filled by a concrete membrane of 1.4 m thickness. The membrane is only 0.7 m on the side supports. A cut of the bridge to emphasize the inner membranes is shown in figure 5.2.

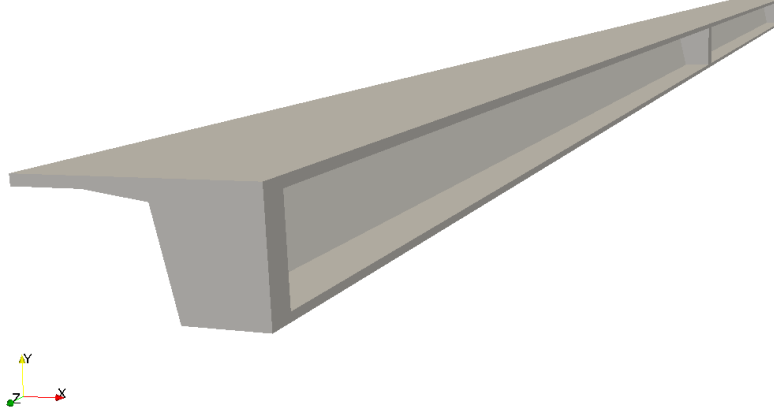


Figure 5.2: Cut section of the bridge.

All the passive reinforcements present in the bridge are  $\phi 16$  reinforcements (diameter 16 mm). The transversal reinforcements are spaced by 20 cm. The spacing for longitudinal reinforcements varies from 18 cm to 20 cm. The reinforcement layout is shown in figure 5.3.

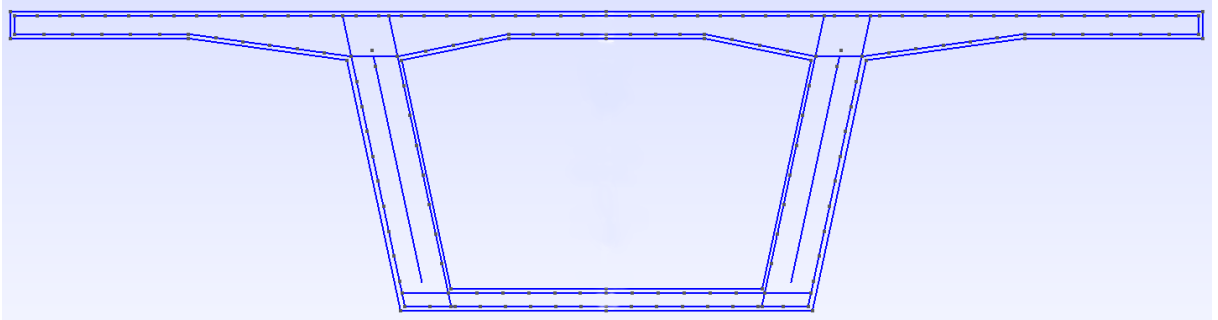


Figure 5.3: Reinforcement layout. Blue lines in the web are post-tension cables. Cover concrete is 40 mm.

The cables for post-tension are made of Y1770S7 steel, in the form of 13  $\phi 15.7$  tendons. There are eight cables in total. The cables form three parabolas. Two large ones have a positive curvature (disposed in the spans of the bridge), and a small one with a negative curvature on the central support, as shown in figure 5.4.

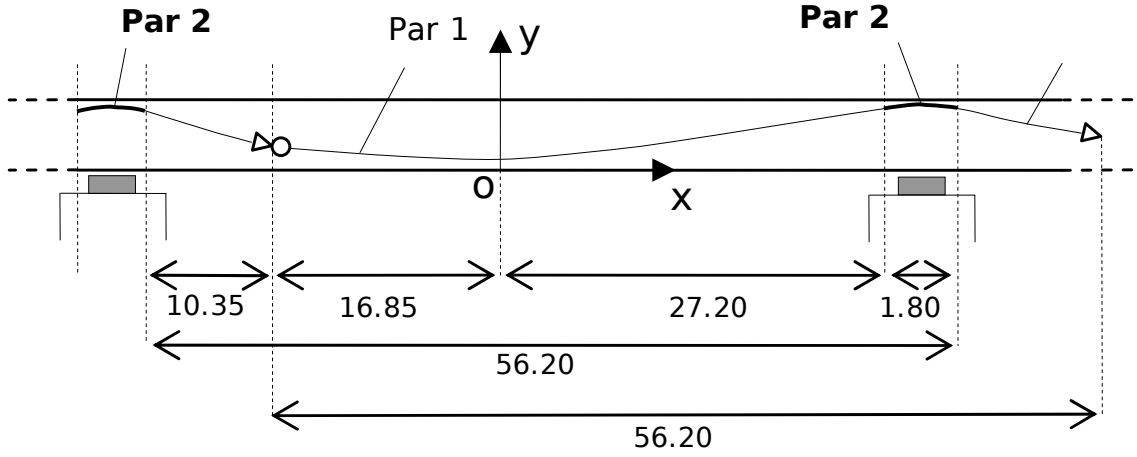


Figure 5.4: Layout cable in a central span of the bridge. Dimensions are in meters. Figure taken from [15].

Table 5.1: Material properties.

	Concrete	Reinforcement	Cable
E	30 GPa	205 GPa	195 GPa
$\nu$	0	-	-
$\rho$	2500 kg/m <sup>3</sup>	-	-
A	-	$2.01 \cdot 10^{-4} \text{ m}^2$	$1.95 \cdot 10^{-3} \text{ m}^2$
$\sigma_p$	-	-	923 MPa

Here, a reduced  $\sigma_p$  is used to take into account the loss of prestress due to friction, relaxation, shrinkage and creep. The loads applied are the self-weight ( $g = 9.81 \text{ m/s}^2$ ), a uniform load of  $2.4 \text{ kN/m}^2$  for the roadway material, punctual loads representing the wheels of a truck, as described by [6], and distributed traffic of  $q_k = 2.5 \text{ kN/m}^2$ . Traffic loads are shown in figure 5.5.





the code. The size of the mesh is the engineer's choice, and very fine meshes would require super-computer clusters to compute, which might not always be available to engineers. The computational cost would be even higher if nonlinear constitutive laws were chosen for the concrete or the reinforcements. In total, there are 260964 degrees of freedom in the model, with 1122554 reinforcement elements after the intersection computation.

## 5.2 Fatigue solicitation

In this simulation, we use linear elastic constitutive laws for both the concrete and the reinforcements. In order to compute results that would make sense in the case of a real bridge, we have to limit the loads applied to the service loads. Characteristic values are directly available in the SIA 261, and we use no partial security factor. The goal here is to compute the fatigue solicitation, which is the difference between the highest and lowest stress in a reinforcement as the truck loads are moving along the bridge. This fatigue solicitation, noted  $\Delta\sigma$  is then compared to values given in the table 13 of SIA 262 in order to verify the fatigue resistance.

The computation of  $\Delta\sigma$  is quite straightforward. The dead loads are first applied to the bridge, and the stress state of the reinforcements is stored. Then the traffic loads are applied at one end of the bridge and start moving along the bridge. For each load position, the maximum and minimum stress a reinforcement element has experienced is recorded. When the load reaches the other side of the bridge, the fatigue solicitation is computed as the difference between the maximum and minimum stress.

It is to be noted that because of the linear elastic constitutive law used for concrete, some of the results obtained are not realistic. Parts of the bridge can be subjected to high  $\Delta\sigma$  because the service loads can cause cracking, which will increase the stress in the reinforcements. This effect will not be correctly captured with a linear elastic constitutive law for concrete. However, we can derive good results for  $\Delta\sigma$  in the post-tension cables, since the prestressing is designed to keep the entire cross-section intact under service loads.

## 5.3 Results

### 5.3.1 Dead loads

Figure 5.7 shows  $\sigma_{zz}$ , the longitudinal stress in the bridge under dead load conditions.

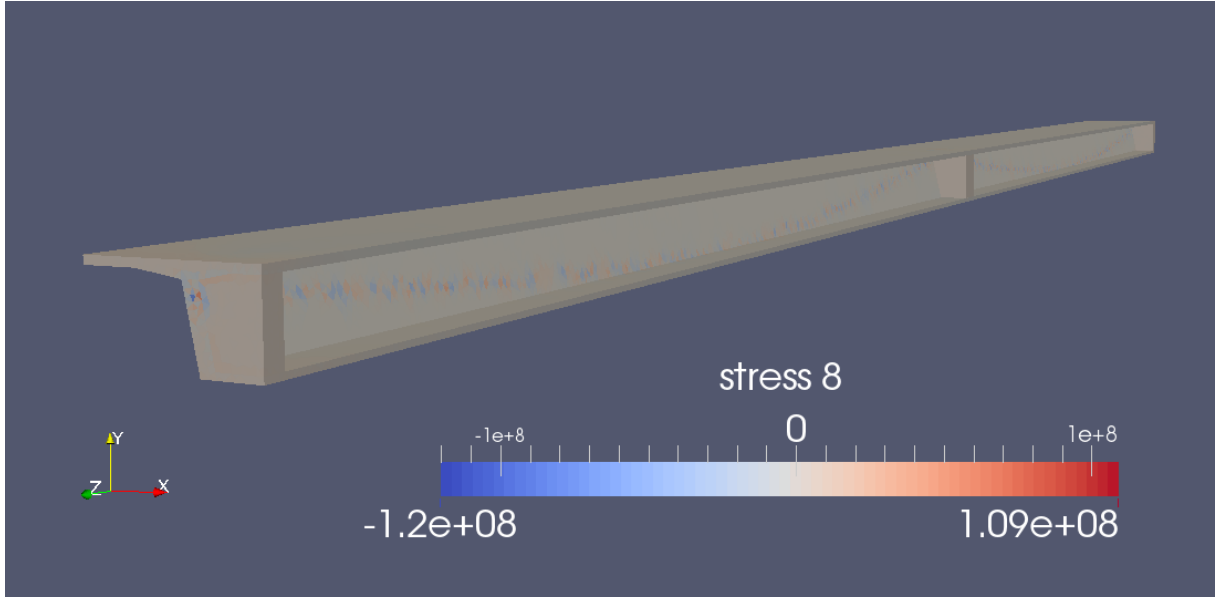


Figure 5.7: Longitudinal stresses under dead loads (cut view). Units are pascals.

Figure 5.7 shows that most of the bridge is subjected to very low longitudinal stresses, because the post-tension compensates the dead loads. There are tension stresses as high as 109 MPa in the concrete, but this is due to the post-tension cables creating a stress concentration on the extremities of the bridge. The path of the cables can be seen in the figure as some elements have higher stresses than the rest of the bridge. This is explained by the curvature of the cable, which creates deviation forces that have to be introduced in the concrete.

Since this model does not take cracking of the concrete into account, it is difficult to interpret and assess the results, especially at the extremities of the bridge where the post-tension cables are anchored. The proper analysis of this zone would require the addition of steel plates for the diffusion of the prestressing force, spiral reinforcements for the concrete confinement, as well as the use of a nonlinear constitutive law for the concrete, which can be difficult to express in 3D.

### 5.3.2 Fatigue

As explained in section 5.2, the fatigue analysis will be limited to the post-tension cables. A plot of  $\Delta\sigma$  along the bridge for both cables is given in figure 5.8.

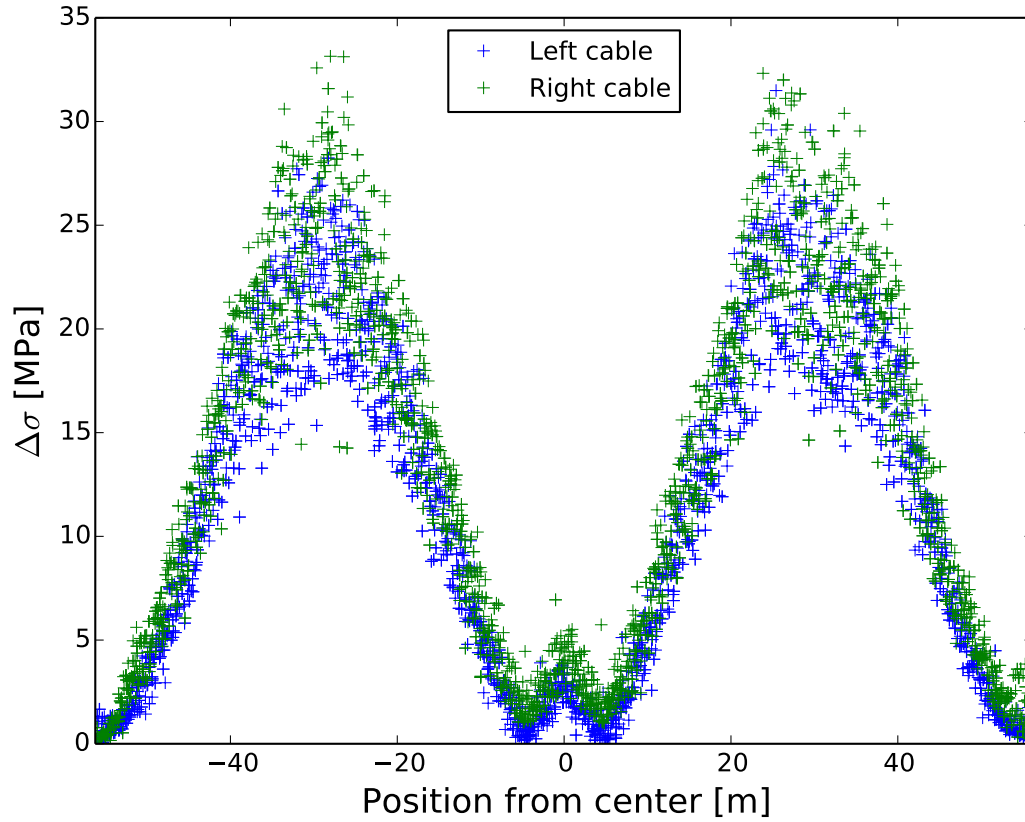


Figure 5.8:  $\Delta\sigma$  for the post-tension cables. Right cable is under the truck loads.

The maximum  $\Delta\sigma$  is given at mid-span for the right cable, for a value of 33.1 MPa, which is under the SIA 292 prescription of  $0.8\Delta\sigma_{sd, \text{fat}} = 0.8 \cdot 145 = 116$  MPa. It can be noted that the right cable, which is the cable under the truck loads, has higher  $\Delta\sigma$  values than the other cable, since the eccentricity of the truck loads cause a torsion in the bridge. The noise phenomenon exhibited in 4.3 is clearly visible in figure 5.8, since the mesh is relatively coarse. It makes the distinction of a clear line difficult, so it is likely that the maximum fatigue stress is over-estimated.

It is interesting to note that the fatigue stress is close to zero at 5 m away from central support on each side. This corresponds to the point where the post-tension cables cross the section's neutral axis, as the axial strain is zero under any load. It is then interesting to place the coupling devices for the cables around this point, as they have lower  $\Delta\sigma$  limits than the rest of the cables (70 MPa instead of 145 MPa).

# Chapter 6

## Preliminary nonlinear analysis

The aim of this section is to show that with the code architecture described in appendix A, the embedded can be used in conjunction with nonlinear constitutive laws with minimum effort. In this section, we qualitatively observe the compatibility of the embedded model with nonlinear constitutive laws for concrete and steel. The study of the nonlinear behavior of concrete is very complex, both from an experimental and a numerical point of view, and this section is not a study of that behavior, neither of the numerical models that approximate it. The example presented is a tension simulation on a sample of reinforced concrete with the following constitutive laws :

- Reinforcements : an elastic-perfectly plastic model (appendix B.1)
- Concrete : a regularized sequentially linear saw-tooth softening model (appendix B.2)

Neither of these constitutive laws were implemented during the project. The goal here is to analyze the response of the reinforced concrete sample in a force-displacement diagram and to compare this response to known behaviors. This will help assess the ability of the embedded model to be coupled with nonlinear constitutive laws.

### 6.1 Model description

The analyzed sample is a cube of concrete of  $1\text{ m} \times 1\text{ m} \times 1\text{ m}$ . Though the dimensions are not those of a real tension experiment, the goal of this simulation is to qualitatively study the behavior of the embedded model with nonlinear material laws. The cube shape is chosen to reduce the number of elements, as non-linear models are more costly than linear elastic models. There are four parallel reinforcements in the cross sections, shown in figure 6.1.

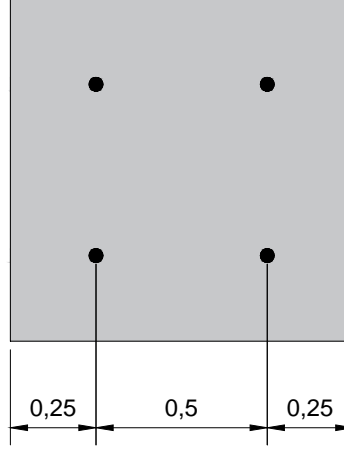


Figure 6.1: Cross-section of reinforced concrete sample.

The material properties of the model are the following :

Table 6.1: Material properties.

	Concrete	Reinforcement
E	30 GPa	205 GPa
$\nu$	0	-
Strength	$f_{ct,m} = 2.9$ MPa	$f_{s,k} = 500$ MPa

In order to avoid damaging all the concrete elements at the same time, we define a random distribution of  $f_{ct}$ . We add a term that follows a uniform distribution between  $[-0.5, 0.5]$  MPa.

The quantity of reinforcement is defined by the reinforcement ratio  $\rho$  :

$$\rho = \frac{A_s}{A_c} \quad (6.1)$$

It is then possible to define the minimum reinforcement quantity, which corresponds to the amount of reinforcements needed to carry the load causing the first crack formation, i.e. when the concrete reaches  $f_{ct}$ .

$$\begin{aligned} F_{c,crack} &= F_s \\ f_{ct,m} A_c &= f_{s,k} A_{s,min} \\ \frac{A_{s,min}}{A_c} &= \frac{f_{s,k}}{f_{ct,m}} = \rho_{min} = 0.58\% \end{aligned}$$

The reinforcement ratio is varied with values shown in table 6.2. The ultimate force is  $F_u = f_{s,k} A_c \rho$  :

Table 6.2: Ultimate forces

$\rho$	$F_u$
0.2%	1000 kN
0.58%	2900 kN
1%	5000 kN
2%	10000 kN
5%	25000 kN

The load is applied on the system as a displacement increment. Three faces of the cube are blocked in their respective normal displacements. The non-blocked face normal to the reinforcements is displaced by  $5 \cdot 10^{-6}$  m per step, as shown in figure 6.2. The force is computed on the blocked face normal to the reinforcements by adding the residuals of the nodes of that face.

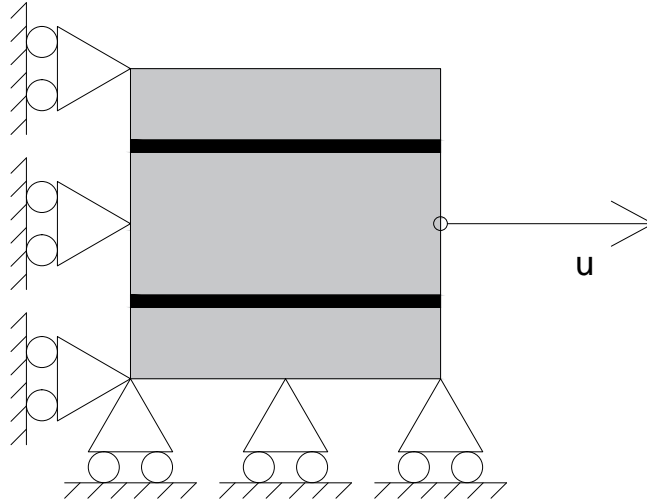


Figure 6.2: Boundary conditions for reinforced concrete sample.  $u$  is imposed on the entire face.

## 6.2 Result analysis

### 6.2.1 Force displacement behavior

The force displacements diagrams are shown in figure 6.3.

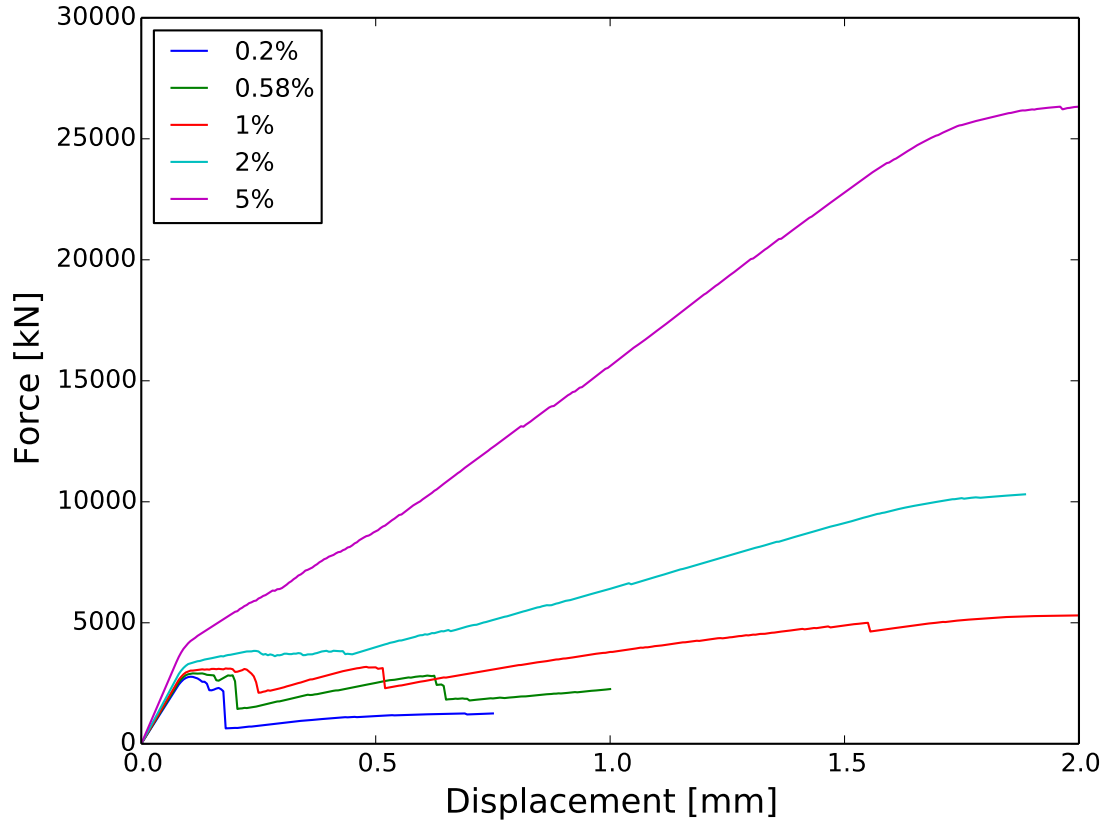


Figure 6.3: Force-displacement diagrams for varying reinforcement ratios.

On figure 6.3, the last point of the curve corresponds to the yielding of the reinforcements. The Newton-Raphson method used in the simulation could not reach equilibrium within the maximum number of iterations once the reinforcement was yielded.

The curves of reinforcement ratios  $\geq 1\%$  can be decomposed in 4 parts, best seen in the 2% curve.



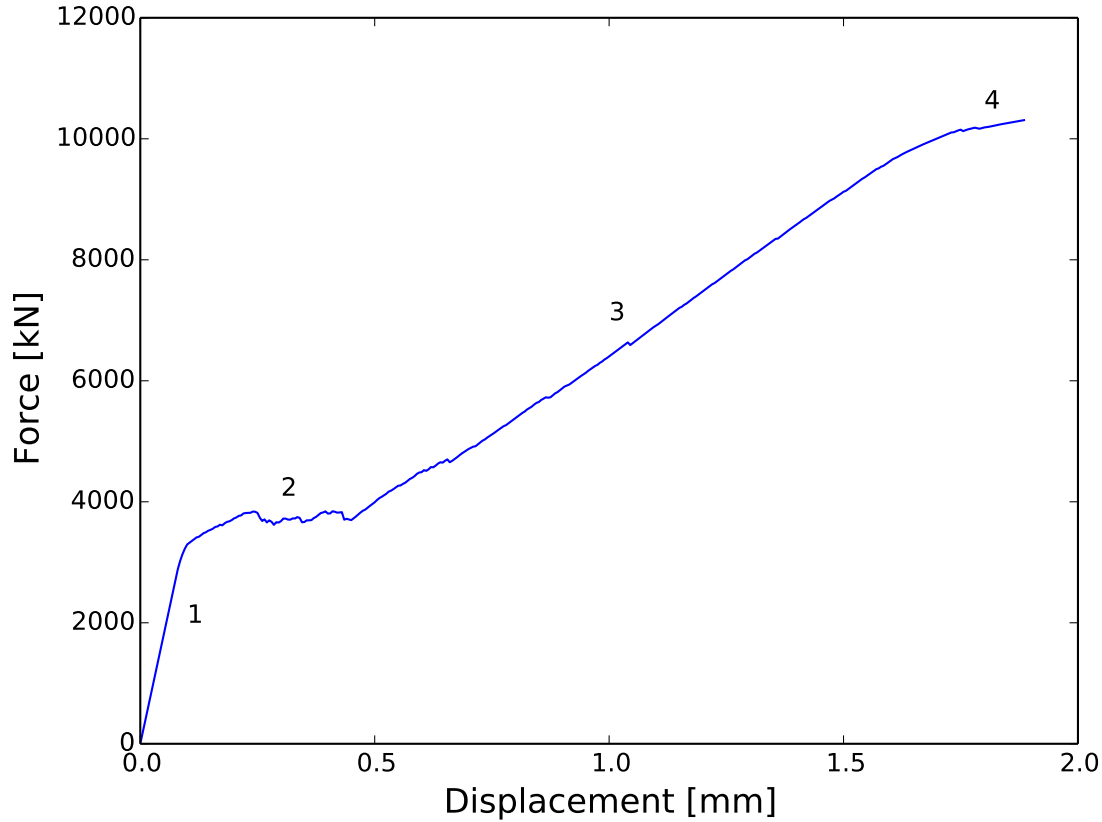


Figure 6.4: Four phases of tension test.  $\rho = 2\%$

These four phases correspond to :

1. Elastic behavior of concrete and reinforcement
2. Tension limit is reached for concrete, damage starts to propagate
3. Damage reaches a point where most of the load is carried by the reinforcement
4. Reinforcement yields

For the  $\rho = 5\%$  curve, phase 2 is almost non-existent. This is explained by the higher stiffness of the reinforcements (proportional to  $A_s$ ), which causes the damage in concrete to spread faster due to the higher loads. The opposite is visible for  $\rho = 1\%$ , where some saw-tooth behavior can still be seen during phase 3 of the curve because the low load is not enough to fully damage the concrete. It can also be noted that the slope of the elastic part increases as  $\rho$  increases. This effect is barely visible for  $\rho \leq 2\%$ , but has to be accounted for at  $\rho = 5\%$ . The curves of figure 6.3 show that the yielding plateau of phase 4 does not match the yielding forced computed in table 6.2. This is not on the safe side of design, since the ultimate resistance is over-estimated. However, this phenomenon can be explained by the fact that the Newton-Raphson method fails to reach equilibrium after yielding of the reinforcements.

For reinforcement ratios below the minimum reinforcement ratio, the behavior of the curve is different. For  $\rho = 0.2\%$ , the amount of reinforcement is not enough to carry the load after the concrete reaches the tensile strength. This is shown by a large drop in the curve (phase 1 in figure 6.5). Then the load is carried by the reinforcements in phase 2. At some point, the reinforcements yield, and the load is carried by concrete again (phase 3). This last phase does not correspond to the real-life behavior of reinforced concrete, and comes from the fact that a residual stiffness is necessary in the concrete to obtain convergence when it is damaged. Practically, a maximum damage is defined smaller than 1 (in this case 0.999). In a real experiment, phase 3 would correspond to the yield plateau of the reinforcements.

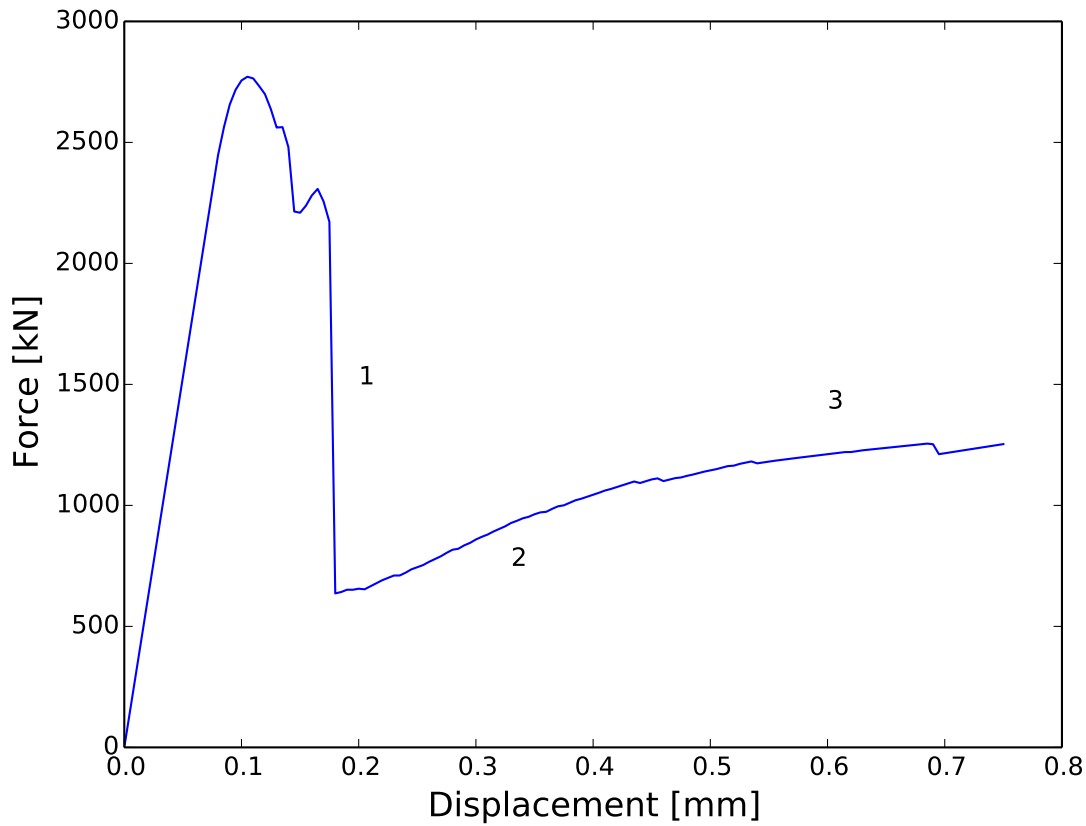


Figure 6.5: Nonlinear phases of tension test.  $\rho = 0.2\%$

In a force controlled experiment, the failure of the sample would occur right after the concrete has cracked if  $\rho < \rho_{\min}$ , and at the reinforcement yield plateau if  $\rho > \rho_{\min}$ , assuming an elastic-perfectly plastic behavior of the reinforcements.

These results show that the embedded model is able to produce consistent results when used in combination of a damage model and a plasticity law. However, the combination of the two non-linearities within the embedded model makes convergence difficult after the yielding of the reinforcement, which is something that could be studied in a further project.

### 6.2.2 Damage propagation and cracks

Although the mesh is relatively coarse for computational cost reasons, it is possible to qualitatively assess the behavior of concrete. As mentioned at the beginning of this chapter, the detailed analysis of the concrete behavior is out of the scope of this project. Figure 6.6 shows the displacement of the cube.

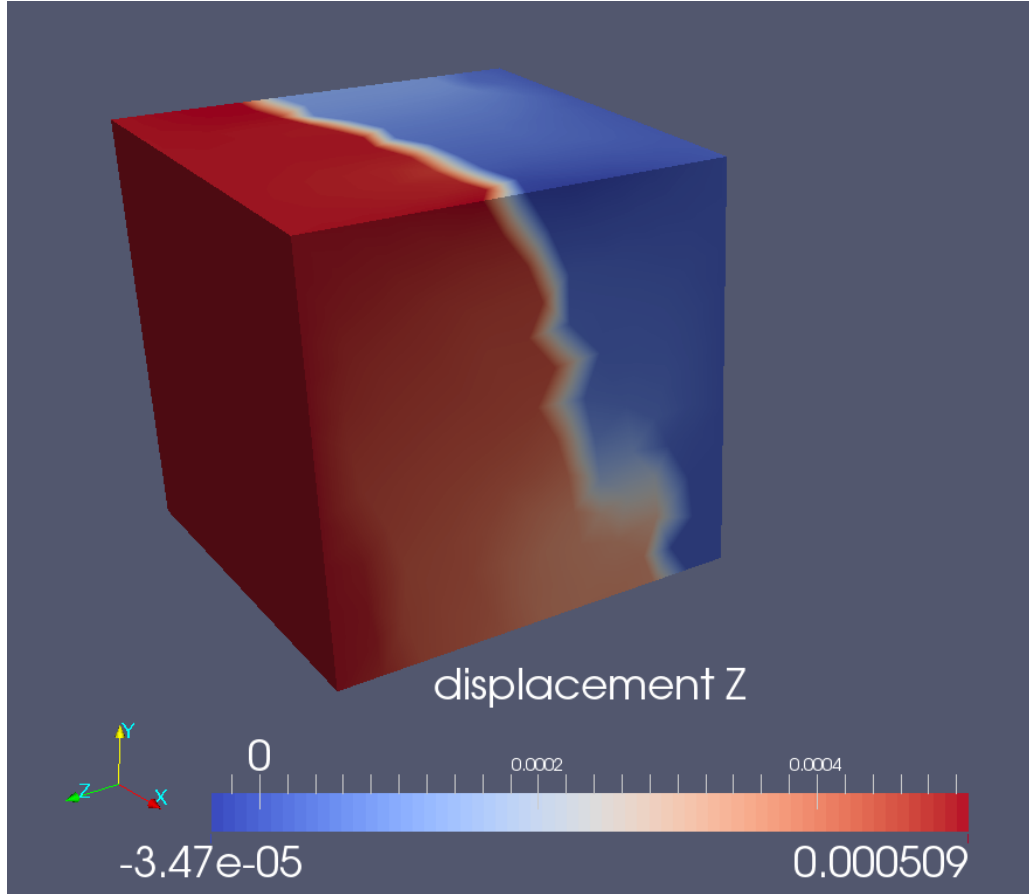


Figure 6.6: Displacement along z-axis in cube for imposed displacement of 0.5 mm. Units are meters.  $\rho = 2\%$

On this figure, a main crack is clearly visible at the middle of the cube. There is a rows of fully damaged elements along the crack path, which causes the displacement discontinuity. The next figure shows the damage inside a cross-section of the cube.

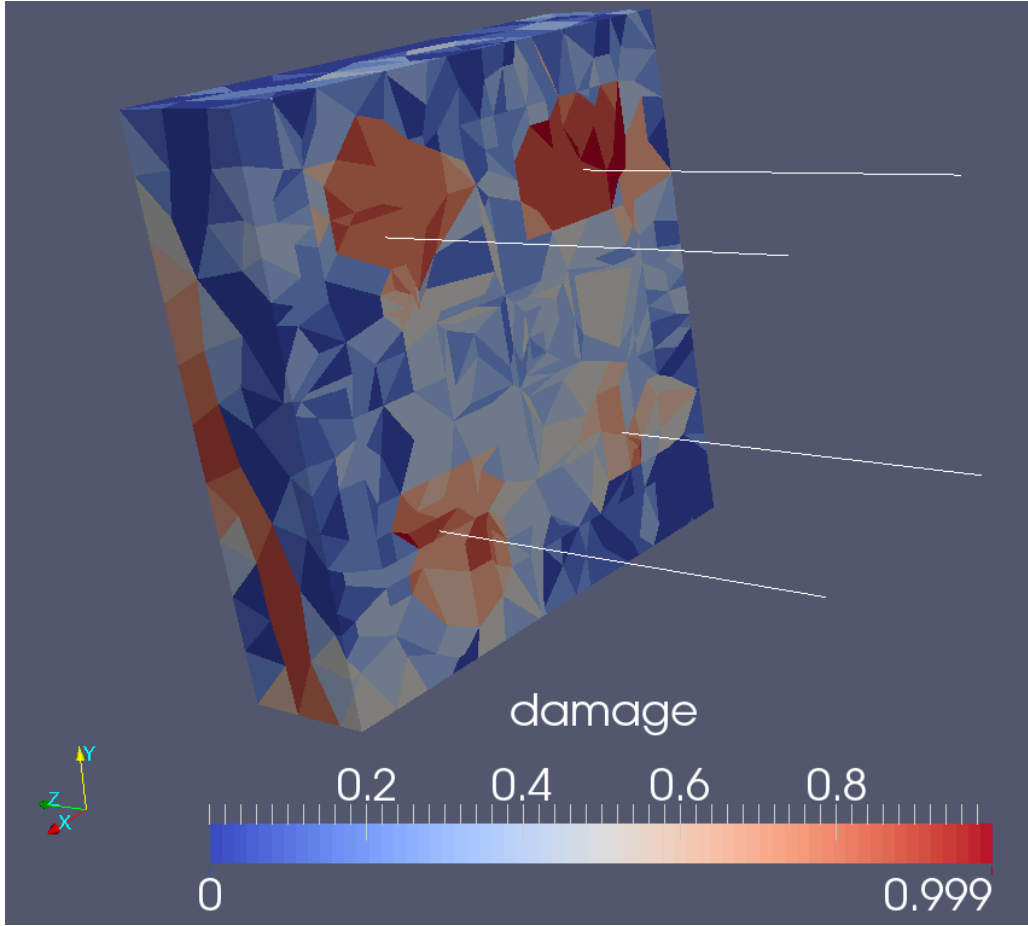


Figure 6.7: Damage in cube for imposed displacement of 0.5 mm. Cut is done at  $z = 0.75$  m.

The damage is clearly localized around the position of the reinforcements (shown as white lines on figure 6.7). This can be explained because of the additional stiffness the reinforcement provides to a series of elements. Since there is a perfect bond between the reinforcements and the concrete, these elements are carrying the load in the reinforcements (corresponding to phase 3 of figure 6.4), and cannot be relieved of stress with the apparition of a crack in the bulk. They are thus taking higher load and are more damaged than the rest of the concrete. This effect would probably be lessened if an imperfect bond was considered for the reinforcements.

# Conclusion

The embedded model is a way to model the contribution of the reinforcement in a reinforced concrete structure that allows for an easier meshing process. Some structures in civil engineering are too complex, with too many reinforcements to be modeled using the discrete model. The use of the embedded model for such structures is necessary, as shown in chapter 5, where the analysis of a real-life bridge was done. However, this flexibility comes at the cost of performance, as it requires finer meshes than the discrete model to provide the same level of precision (chapter 3), and exhibits a noise phenomenon for coarse meshes (chapter 4). But the drawbacks can be removed in all the cases where the use of the discrete model is possible, because the embedded model is a generalization of the discrete model (chapter 1).

The embedded model also provides flexibility in terms of constitutive laws, since any constitutive tensor  $\mathbf{D}$  can be used in the integration of the stiffness matrix. A good code design can allow the user to swap constitutive laws for the reinforcements (appendix A). It is also compatible with non-linear constitutive laws for the concrete and gives results coherent with known behavior of reinforced concrete in tension (chapter 6). The use of an efficient intersection algorithm improves the performance of the model (chapter 2), making it usable for complex structures on a personal computer. It was used in a prestressed concrete bridge to analyze the fatigue solicitation on the post-tension cables under road traffic loads.

## Personal aspects

The project was very interesting for me, as I improved greatly my knowledge of the finite elements, reinforced concrete and C++. Working in an open-source project like Akantu is very enriching, and discussions with developers are always stimulating, as well as the discussions of the results obtained through the simulations. Although there are aspects of the embedded model that I could not study in details, both because of a lack of time and theoretical background (like convergence issues), these aspects can be studied in further projects.

## Acknowledgments

The author wishes to thank Mr. Marco Vocialta and Dr. Mauro Corrado for their helpful insight on the subjects treated in this report, Dr. Nicolas Richart and Dr. Guillaume Anciaux for their valuable knowledge of C++, Akantu and finite elements, Ms. Aurelia Cuba-Ramos for her expertise of the concrete damage model used in this project, and

Prof. Molinari for the opportunity of working on such a project. Special thanks also go to my parents, Sarah Anderson, Benjamin Paccaud and the LSMS for their great support.

# Appendix A

## Computational geometry, code organization and algorithms

This appendix describes the organization of the code implementing the embedded model in the finite elements library Akantu. It is divided into two parts. The first part refers to additions into Akantu's solid mechanics features. The second refers to a set of computational geometry features implemented in Akantu that are not directly related to the embedded model.

### A.1 Embedded model and materials

Implementing the embedded model in Akantu's solid mechanics model required the design of new classes. First of all, from equation (1.4), some quantities have to be evaluated on the quadrature points of the embedded element, then integrated along the latter. In Akantu, the easiest way to do this is to use the `FEEngine` class defined on a mesh that represents the embedded elements. This class will handle the integration and assembly of the stiffness matrix. Note that since equation (1.4) needs to be integrated along the reinforcement, but  $\mathbf{K}_{s,i}$  needs to be assembled on the bulk nodes, two `FEEngine` objects will be used, as well as two meshes. Since the base class `SolidMechanicsModel` in Akantu does not handle these operations, a new model was created : the `EmbeddedInterfaceModel`, which inherits from `SolidMechanicsModel`.

The actual computation of the stiffness matrix is done in the class `Matrial`. It computes, integrates and assembles  $\int \mathbf{B}^T \mathbf{D} \mathbf{B} dV$ , with the help of `FEEngine`. However, the embedded model requires the computation of  $\int \mathbf{B}^T \mathbf{C}^T \mathbf{D} \mathbf{C} \mathbf{B} ds$ . So a new class was created, `MaterialReinforcement`, that redefines the `assembleStiffnessMatrix()` method, as well as the `updateResidual()` method, since the computation of the residuals in the embedded elements is different from the bulk elements.

In addition, `MaterialReinforcement` defines an important method to link the reinforcement mesh and the concrete mesh : `filterInterfaceBackgroundElements()`, which returns the element ids of the concrete elements that are related to the reinforcement elements. This allows the default `FEEngine`, defined on the concrete mesh, to assemble  $\mathbf{K}_{s,i}$  into the global stiffness matrix.

The last thing to do is to compute  $\mathbf{D}$  and  $\sigma$  according to the chosen constitutive law. There are already many constitutive law implemented in Akantu, and it would

be a waste of time not to use them. So a second new material has been added : `MaterialReinforcementTemplate`, which inherits from `MaterialReinforcement` and a template parameter called `ConstLaw`. This allows the user to easily switch constitutive laws without having to implement a new class for each supported constitutive law. The following diagram summarizes the new classes and their role in the solid mechanics branch of Akantu.

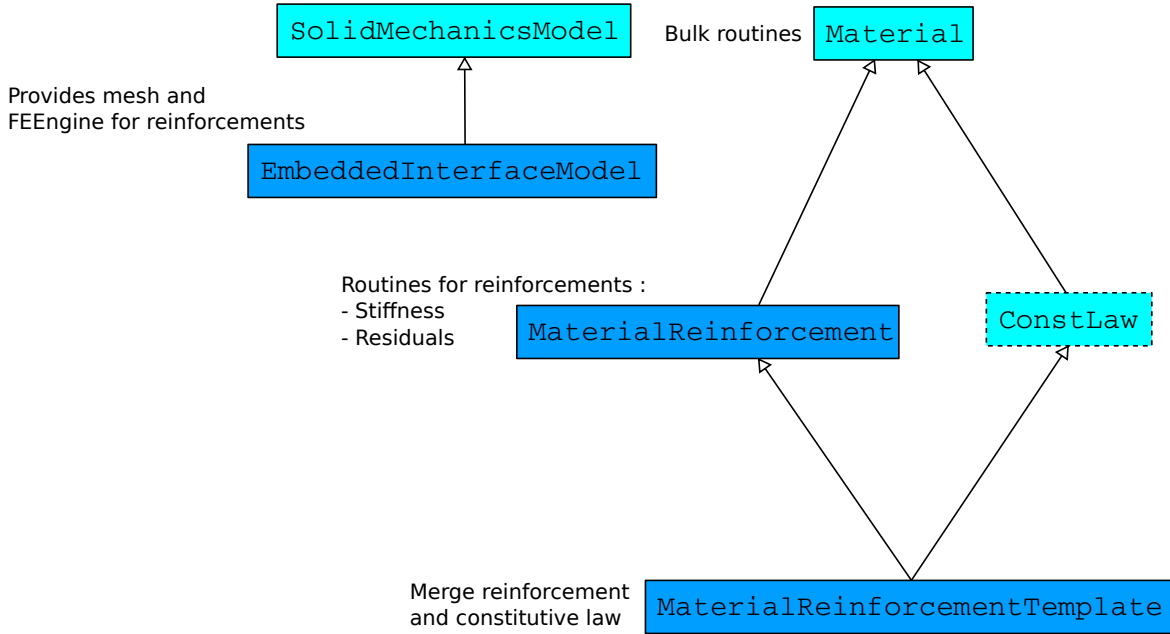


Figure A.1: Inheritance diagram for the new classes in solid mechanics module

## A.2 Mesh geometry module

The mesh geometry module is a new feature in Akantu that was developed during the project. It is meant as a bridge between Akantu and the Computational Geometry Algorithms Library (CGAL). Since CGAL has very high level algorithms, it was decided that the mesh geometry module would not be redefining CGAL's object in Akantu. This means that CGAL is necessary to make the mesh geometry module work, and that if one day Akantu's developers want to stop using CGAL, they will have to rewrite the module almost entirely. But the gain from this strong dependency is that it becomes easier to use CGAL's powerful algorithms.

The only algorithm currently used in Akantu is the AABB-tree algorithm [7]. It is used for the embedded model and might be used in the future in Akantu's other features. The integration of the algorithm into Akantu is done through two main classes :

- `MeshGeomFactory`, which will help make the AABB tree
- `MehsGeomIntersector`, which will compute the intersections and create the necessary data



**MeshGeomFactory** decomposes elements of a mesh into **Primitive** objects. The **Primitive** objects can be of any type on which CGAL can compute an intersection. For example, a 2D mesh of triangles will be processed by **MeshGeomFactory** into a list of triangles. A 3D mesh of tetrahedrons will however be processed into triangles, because CGAL can't use tetrahedrons to compute intersections. Each tetrahedron is therefore decomposed into its faces, which will make up the **Primitive** objects of the mesh.

The **MeshGeomIntersector** class is a base class for several “intersector” classes. Each “intersector” defines a **Query** object. In the case of **MeshSegmentIntersector**, the **Query** object is a segment. The “intersector” implements an algorithm that creates the data needed by the model to do the finite elements computations. In the case of the embedded model, it is necessary to construct a mesh of the reinforcement where each node corresponds to the intersection of a reinforcement and the bulk elements of the model. The following diagram shows the main classes and their role in the mesh geometry module. For more information on the module, please refer to the Akantu developer manual (<https://lsmssrv1.epfl.ch/trac-akantu/wiki>).

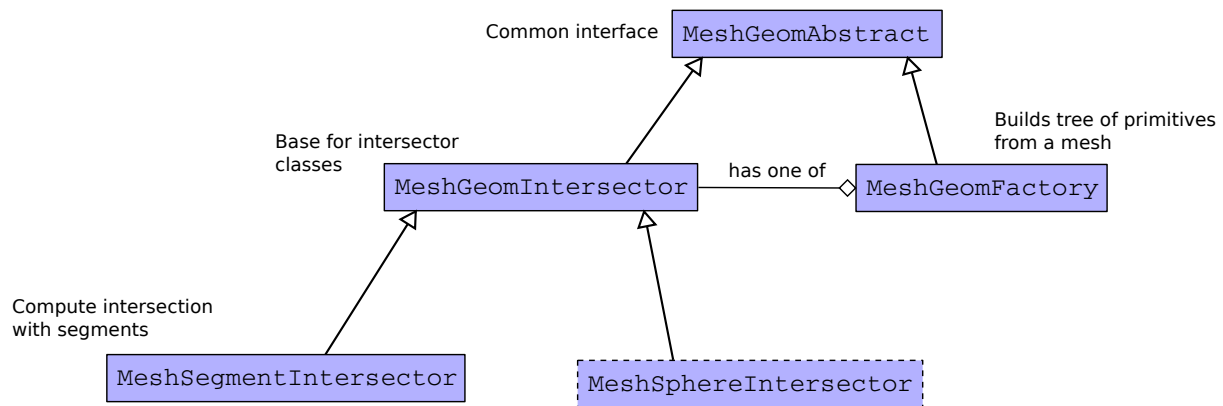


Figure A.2: Comprehension diagram for the mesh geometry module

# Appendix B

## Constitutive laws and algorithms

This section describes the constitutive laws and numerical algorithms used in the project to model non-linear behavior in steel and concrete.

### B.1 Isotropic linear hardening

Even though a linear elastic constitutive law was used for most of the simulations in the project, a constitutive behavior was described in section 1.2.1 for passive reinforcements. In the last simulation, an idealized behavior was used, with the assumption that the reinforcement was behaving like an elastic-perfectly plastic solid, with the following one dimensional stress-strain curve :

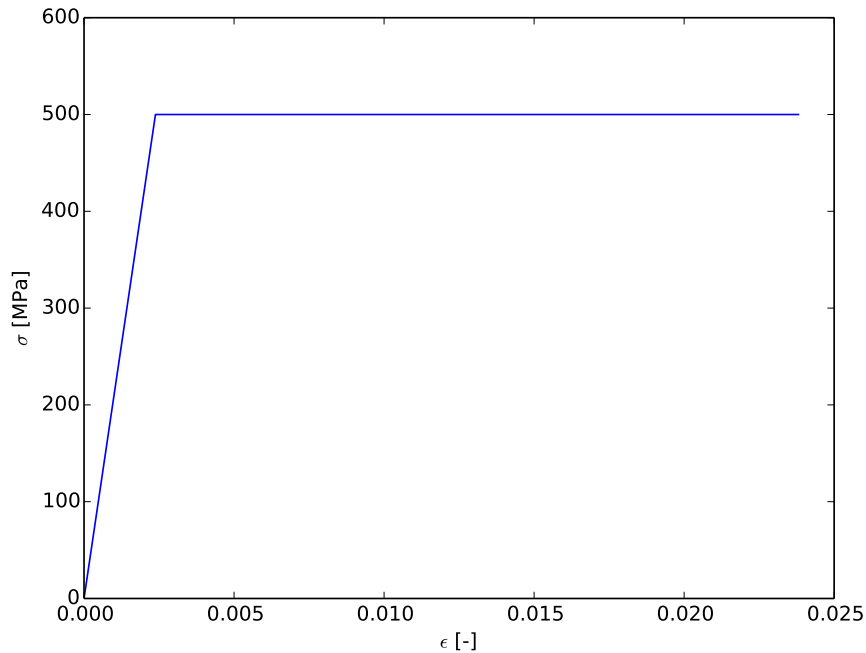


Figure B.1: Elastic-perfectly plastic behavior.

This behavior can be generalized in an rate-independent isotropic linear hardening, constitutive law. Here the hardening modulus is zero. The algorithm used in

Akantu to reproduce this constitutive law is the following (from [10], modified for thermal stresses [12]) :

1. Compute trial stress :

$$\sigma_{ij}^{\text{tr}} = \lambda \Delta \varepsilon_{kk} \delta_{ij} + 2\mu \Delta \varepsilon_{ij} + \Delta \sigma_{ij}^{\text{th}} + \sigma_{ij}^{\text{cur}} \quad (\text{B.1})$$

where  $\Delta \varepsilon_{ij}$  is the strain increment,  $\sigma_{ij}^{\text{cur}}$  is the current stress state and  $\sigma_{ij}^{\text{th}}$  is the thermal stress.

2. Compute Von Mises stress :

$$s_{ij} = \sigma_{ij}^{\text{tr}} - \frac{\sigma_{kk}^{\text{tr}}}{3} \quad (\text{B.2})$$

$$\sigma^* = \sqrt{\frac{3}{2} s_{ij} s_{ij}} \quad (\text{B.3})$$

where  $s_{ij}$  is the deviatoric part of  $\sigma^{\text{tr}}$  and  $\sigma^*$  is the Von Mises stress.

3. If there is yielding, compute the isotropic hardening  $r$  and the plastic multiplier  $\Delta p$  using the radial return implicit method.

```

 $\Delta p \leftarrow 0$ 
 $d\Delta p \leftarrow 0$ 
while  $\sigma^* - r - \sigma_y \geq 0$  do
   $d\Delta p \leftarrow \frac{\sigma^* - 3\mu \cdot \Delta p - r - \sigma_y}{3\mu + h}$ 
   $\Delta p \leftarrow \Delta p + d\Delta p$ 
   $r \leftarrow r + h \cdot d\Delta p$ 
end while

```

where  $\sigma_y$  is the yield limit,  $\mu$  is the shear modulus and  $h$  the hardening modulus.

4. Compute the plastic strain :

$$\Delta \varepsilon_{ij}^{\text{p}} = \frac{3}{2} \Delta p \frac{s_{ij}}{\sigma^*} \quad (\text{B.4})$$

5. Compute the elastic strain :

$$\Delta \varepsilon_{ij}^{\text{e}} + \Delta \varepsilon_{ij}^{\text{th}} = \Delta \varepsilon_{ij} - \Delta \varepsilon_{ij}^{\text{p}} \quad (\text{B.5})$$

6. Compute the stress increment :

$$\Delta \sigma_{ij} = \lambda \Delta \varepsilon_{kk}^{\text{e}} \delta_{ij} + 2\mu \Delta \varepsilon_{ij}^{\text{e}} + \Delta \sigma_{ij}^{\text{th}} \quad (\text{B.6})$$

With the stress increment, it is possible to compute the residuals. The latter are then used in the Newton-Raphson method to converge to a solution that guarantees equilibrium within a certain tolerance.

## B.2 Regularized sequentially linear saw-tooth softening

This section describes the model used in the non-linear simulation to represent the response of concrete in a tension test. The model, described in [17], is a damage model that uses linear secant approximations to model the response of the concrete. This produces a saw-tooth behavior : each time an element reaches a critical stress (the tensile strength), its elastic modulus is reduced. The procedure implemented in Akantu is the following :

1. Perform linear analysis on model
2. Find the critical gauss point, i.e. the gauss point with highest normalized stress  $\sigma^* = \frac{\sigma_I}{f_t}$  where  $\sigma_I$  is the maximum eigenvalue of  $\sigma$
3. Reduce stiffness of critical gauss point if damage condition  $\sigma^* > 1$  is satisfied. In this project, the stiffness is reduced by 10% every step
4. Repeat until no gauss point satisfies the damage condition

The reduction of the stiffness is done by a damage parameter  $d$  which can take values between 0 (no damage) and a maximum, here set to 0.999 for numerical reasons exposed in 6.2.1. At each step, a gauss point satisfying the damage condition sees its damage parameter increase by 0.1. The maximum number of damage steps is 8, at which point the gauss point is considered fully damaged, and  $d$  is set to 0.999. The stress is computed as :

$$\sigma = (1 - d)C_e \varepsilon \quad (\text{B.7})$$

Where  $C_e$  is the elasticity tensor. This model presents the advantage that it “always provides a solution: the secant saw-tooth stiffness is always positive, so that ill-conditioning or divergence does not appear in sequentially linear analysis.” Another advantage is, as stated in [17] :

A physical explanation to the model is that fracture is a gradual separation process whereby the net cross section that connects material, and thus the stiffness, is gradually reduced. An advantage of the model is that the regular notions of fracture mechanics, like the principal tensile stress criterion, the envelope strength and fracture energy are maintained which helps in reaching realistic energy consumption and toughness as observed in experiments.

Rots and Invernizzi also mention problems that arise with this model, such as mesh dependency. They provide solutions to those problems, but these were not implemented in the model that was used for the nonlinear simulation.

# Bibliography

- [1] Akantu. <http://lsms.epfl.ch/akantu>.
- [2] Akantu user's guide. [http://lsms.epfl.ch/files/content/sites/lsms/files/akantu\\_ug\\_v2.2.pdf](http://lsms.epfl.ch/files/content/sites/lsms/files/akantu_ug_v2.2.pdf).
- [3] Cgal. <http://www.cgal.org/>.
- [4] jconc. [http://i-concrete.epfl.ch/aide/iconc/iconc01-1-01\\_e.asp](http://i-concrete.epfl.ch/aide/iconc/iconc01-1-01_e.asp).
- [5] *Norme SIA 261*. SIA, 2013.
- [6] *Norme SIA 262*. SIA, 2013.
- [7] ALLIEZ, P., TAYEB, S., AND WORMSER, C. 3D fast intersection and distance computation. In *CGAL User and Reference Manual*, 4.6 ed. CGAL Editorial Board, 2015.
- [8] CHANG, T., TANIGUCHI, H., AND CHEN, W. Nonlinear finite element analysis of reinforced concrete panels. *Journal of Structural Engineering* 113 (1987), 122–140.
- [9] CHENG, Y., AND FAN, Y. Modeling of reinforcement in concrete and reinforcement confinement coefficient. *Finite element analysis and design* 13 (1993), 271–284.
- [10] DUNNE, F., AND PETRINIC, N. *Introduction to computational plasticity*. Oxford Press, 2006.
- [11] FREY, F., AND JIROUSEK, J. *Traité du génie civil, Volume 6. Méthode des éléments finis*. Presses Polytechniques et Universitaires Romandes, 2001.
- [12] FRÉROT, L. Thermo-mechanical coupling of solid mechanics and heat transfer models in akantu. Semester project report, EPFL, 2014.
- [13] GOMES, H., AND AWRUCH, A. Some aspects on three-dimensional numerical modelling of reinforced concrete structures using the finite element method. *Advances in engineering software* 32 (2001), 257–277.
- [14] LESTUZZI, P., AND BADOUX, M. *Génie parasismique: conception et dimensionnement des bâtiments*. PPUR presses polytechniques, 2008.
- [15] MUTTONI, A. Course notes of “ponts en béton”, 2014.
- [16] MUTTONI, A., AND FERNANDEZ, M. Course notes of “structures en béton”, 2014.

- [17] ROTS, J. G., AND INVERNIZZI, S. Regularized sequentially linear saw-tooth softening model. *International Journal for Numerical and Analytical Methods in Geomechanics* 28, 7-8 (2004), 821–856.
- [18] VECCHIO, F. J., AND COLLINS, M. P. The modified compression-field theory for reinforced concrete elements subjected to shear. In *ACI Journal Proceedings* (1986), vol. 83, ACI.