Modeling of the interaction between corrosiondiffusion and mechanical behavior of metallicalloys by FEM coupling

Auteur : Heremans, Julien
Promoteur(s) : Duchene, Laurent; Habraken, Anne
Faculté : Faculté des Sciences appliquées
Diplôme : Master en ingénieur civil des constructions, à finalité spécialisée en "civil engineering"
Année académique : 2018-2019
URI/URL : http://hdl.handle.net/2268.2/6767

Université de Liège - Faculté des Sciences Appliquées

Master thesis submitted in partial fulfillment of the requirements for the degree of Master in Civil Engineering

---

Modeling of the interaction between corrosion diffusion and mechanical behavior of metallic alloys by FEM coupling

---

Julien Heremans

Supervisor:	Laurent Duchene	(**ULiège**)
Co-Supervisor:	Anne Marie Habraken	(**ULiège**)
Jury Members:	Frédéric Collin	(**ULiège**)
	Hélène Morch	(**ULiège**)
	Jocelyn Delahaye	(**ULiège**)
	Manuel Smolders	(**John Cockerill**)

Academic Year 2018-2019

This page is intentionally left blank.

Abstract

This work aimed at the modeling the interaction between corrosion diffusion and the mechanical behavior of metallic alloys, by the development of a coupling finite element method model. A non-fickian diffusion process was considered to interact with a two-dimensional mechanical model, assuming a plane stress state. Both models were coupled in a linear differential system solved by an approached implicit integration scheme.

The validation of the model was performed based on experimental results of metallic alloys submitted to hot corrosion, without mechanical loading. The numerical predictions matched the experimental results with a maximal error reaching 25%.

Corrosion induced damage was modeled by a damage law, modeling a linear degradation of the Young's modulus with the corrosion rate. The diffusion of the corrosion was therefore inducing stress redistribution around the softened material, which in turn increased the peak of hydrostatic pressure. Stress-driven diffusion, whose amplitude was conditioned by a pressure factor, was found to accelerate or decelerate the natural corrosion diffusion process depending on the nature of the stress. This phenomenon was specifically observed in zones submitted to high gradient of hydrostatic pressure.

This page is intentionally left blank.

Acknowledgments

I would first like to thank my thesis supervisor, Prof. Laurent Duchene, for the strong academic support provided during this semester and for his constant availability.

I would also like to thank the people who were not directly implied in this work, but contributed to its accomplishment with their suggestions and discussions: Roman Boman, who was very helpful regarding his suggestions about the computing tools used in this work; Julien Leclerc and Hélène Morch for their sound advice and their constructive comments.

I wish finally to thanks all those who are not explicitly mentioned but who have contributed to the proper accomplishment of this work.

This page is intentionally left blank.

Contents

List of Figures

List of Tables

This page is intentionally left blank.

List of Symbols

Symbol	Description	Unit
A	Cross section of 1D elements	m^2
B_i	First derivative of the shape function associated to degree of freedom i	m^{-1}
$B_{i,j}$	First derivative of the shape function associated to dof. j with respect to variable i	m^{-1}
D	Oxygen diffusivity	$\mathrm{m}^2\,\mathrm{s}^{-1}$
E	Young Modulus	MPa
J	Determinant of the jacobian matrix	m^2 or m
L	Length of the 1D element	m
M	Pressure factor	MPa^{-1}
N_{dof}	Total number of degree of freedom considered in the mesh	–
N_{el}	Total numbers of finite elements considered in the mesh	–
N_i	Shape function associated to degree of freedom i	–
Q	activation enthalpy of oxidation	$\mathrm{kJ\,mol}^{-1}$
R	ideal gaz gas constant	$\mathrm{J\,mol}^{-1}\,\mathrm{K}^{-1}$
T	Temperature / Absolute temperature	°CorK
X	Oxide layer thickness	µm
V	Studied volume of body	m^3
Δm	Mass gain due to oxidation	$\mathrm{kg\,m}^{-2}$
Δt	Time step	s
ε_{xx}	Axial strain in direction x	–
η	Reduced coordinate in local coordinate system (ordinates)	–
γ_{xy}	Shear strain between x and y directions	–
ν	Poisson's ratio	–
ϕ	Damage variable	–
ϕ_c	"Flux of corrosion"	m^{-1}
ρ	Volumetric mass density	$\mathrm{kg\,m}^{-3}$
σ_{xx}	Axial stress in direction x	MPa
τ_{xy}	Shear stress between directions x and y	MPa

Symbol	Description	Unit
ξ	Reduced coordinate in local coordinate system (abscissa)	–
Γ	Surface boundaries	m
Ω	Studied surface	m^2
a	acceleration	$m\,s^{-2}$
c	Relative concentration of oxygen	–
c_{cr}	Critical concentration of oxygen	%
c_{ox}	Threshold concentration of full oxidation	–
k_p	Oxidation growth rate	$kg^2\,m^{-4}\,s^{-1}$
n	Number of Gauss points / Order of Legendre polynomial	–
n_{dof}	Number of degree of freedom in the considered finite element	–
p	Hydrostatic Pressure	MPa
t	time	s
u	Displacement along x	m
v	Displacement along y	m
x	Position in global cartesian coordinate system (abscissa)	m
y	Position in global cartesian coordinate system (ordinates)	m
z	Position in global cartesian coordinate system (elevation)	m
$\mathbf{g}$	Global nodal load vector	Variable
$\mathbf{g}_c$	Nodal load vector (corrosion)	$m^3\,s^{-1}$
$\mathbf{g}_e$	Elementary nodal load vector	Variable
$\mathbf{g}_u$	Nodal load vector (displacements)	N
$\mathbf{q}$	Global nodal unknowns vector	m or –
$\mathbf{q}_c$	Elementary nodal unknowns vector (corrosion)	–
$\mathbf{q}_e$	Elementary nodal unknowns vector	m or –
$\mathbf{q}_\phi$	Damage variable ϕ at nodes	–
$\mathbf{q}_u$	Elementary nodal unknowns vector (displacements)	m
$\mathbf{x}$	Abscissa (expressed in global coordinate system) of the n_{dof} nodes related to the considered element	m
$\mathbf{y}$	Ordinates (expressed in global coordinate system) of the 4 nodes related to the considered element	m
$\boldsymbol{\varepsilon}$	Strain vector	–
$\boldsymbol{\sigma}$	Stress vector	MPa
$\tilde{\boldsymbol{\sigma}}$	Stresses sampled at nodes	MPa
$\boldsymbol{\sigma}^*$	Smoothed stress field	MPa

Symbol	Description	Unit
$\hat{\boldsymbol{\sigma}}$	Stresses sampled at gauss points	MPa
$\mathbf{B}$	First derivative of the shape function matrix with respect to x / Strain shape functions	m^{-1}
$\mathbf{B}_i$	First derivative of the shape function matrix with respect to variable i	m^{-1}
$\mathbf{C}$	Global damping matrix	m^3
$\mathbf{C}_{cc}$	Damping matrix corrosion-corrosion	m^3
$\mathbf{D}$	Constitutive matrix	MPa
$\mathbf{J}$	Jacobian matrix	m
$\mathbf{K}$	Global stiffness matrix	Variable
$\mathbf{K}_{cc}$	Stiffness matrix corrosion-corrosion	$\mathrm{m}^3\,\mathrm{s}^{-1}$
$\mathbf{K}_{cu}$	Stiffness matrix corrosion-displacements	$\mathrm{m}^2\,\mathrm{s}^{-1}$
$\mathbf{K}_e$	Elementary stiffness matrix	Variable
$\mathbf{K}_{uc}$	Stiffness matrix displacements-corrosion	$\mathrm{N\,s\,m}^{-1}$
$\mathbf{K}_{uu}$	Stiffness matrix displacements-displacements	$\mathrm{N\,m}^{-1}$
$\mathbf{L}_x$	First derivative of the strain function matrix $\mathbf{B}$ with respect to x	m^{-2}
$\mathbf{L}_y$	First derivative of the strain function matrix $\mathbf{B}$ with respect to y	m^{-2}
$\mathbf{N}$	Shape function matrix (associated to an element)	–
$\mathbf{N}_c$	Shape function matrix (corrosion problem)	–
$\mathbf{N}_u$	Shape function matrix (mechanical problem)	–
$\mathbf{S}$	strain operator	m^{-1}

This page is intentionally left blank.

Introduction

Context of the study

Numbers of applications in engineering are expected nowadays to achieve progressively higher levels of performance in terms of yields, fuel economy, noise and pollutant emissions, etc. Such challenges require materials to be continually solicited by higher pressures, at higher temperatures, and sometimes also in more hostile chemical environments. Various studies have shown that metal alloys at high temperature were subjected to accelerated corrosion whose effects are increasing with operational temperature. Corrosion induced by aggressive environments such as molten salts, hydroxides, sulfates, etc. are also well known to impact negatively the proper functioning of many engineering systems and hold subsequently a prominent role in material selection.

Readers will have understood that corrosion phenomenon represents a significant stake for such applications as corrosion induced damage, if not taken into account, can lead to unrealistic predictions of the life time of certain pieces of equipment or even worse, to unconservative design of them.

If mechanical and corrosion problems have been investigated independently trough various studies, it exists however few models accounting for both mechanical behavior and corrosion effects. Such coupling models are indeed rarely observed in the literature. [2] studied a mechanical-corrosion coupling with an approach based on cristal-plasticity, while [3] opted rather for a macroscopic approach. The purpose of this study is to develop another macroscopic model using finite element method, dedicated to a coupling model between corrosion and mechanical problems.

The present study takes place more precisely further to the project `Solar Perform` [4] where the nonlinear mechanical behavior of nickel based alloys dedicated to heat exchanger of solar towers. In the frame of this project, corrosion effects have been left apart to focus exclusively on the material behavior (influence of temperature, creep, cyclic loading, etc.). A second project, `Solar Gnext` [5], is starting in August 2019 and is dedicated prediction of corrosion effects for the same applications. The pursued objective is thus to perform a full integration of the effects of corrosion in the previous model. This master thesis constitutes hence an introduction to project `Solar Gnext`.

This study is about numerical modeling and is consequently not a study of the corrosion phenomenon itself. Of course, such numerical development is not practicable without few physical and chemical considerations, especially for validation

purposes.

In the frame of this project, numerical calculations are carried out through a code entirely developed in `Fortran 90`. The choice of the language was motivated by the future integration of the routines in `Lagamine` [6], which is entirely written in `Fortran`.

Targeted applications

As briefly mentioned this master thesis falls within the continuity of SOLAR PERFORM project and as a introduction to `Solar Gnext`. The central application explored by these projects was the design of heat exchanger of solar thermal power plants. The context of this study is presented more in detail in this section in order to highlight the challenges issues tackled by these projects.

Solar thermal power plants are made up of a central tower overhung by a solar receiver where are concentrated all incident solar radiation reflected by heliostats (mirrors). As the entire solar power are concentrated on the solar receiver, it operates at high temperature nearby 700°C and transfers a major part of the incident heat to heat exchangers [7]. While classical solar power plants use water as heat transfer fluid, other projects [8] have used molten salts instead (water was still used in secondary circuit for steam production). Figure 1 shows a simplified functioning of a such solar power plant.

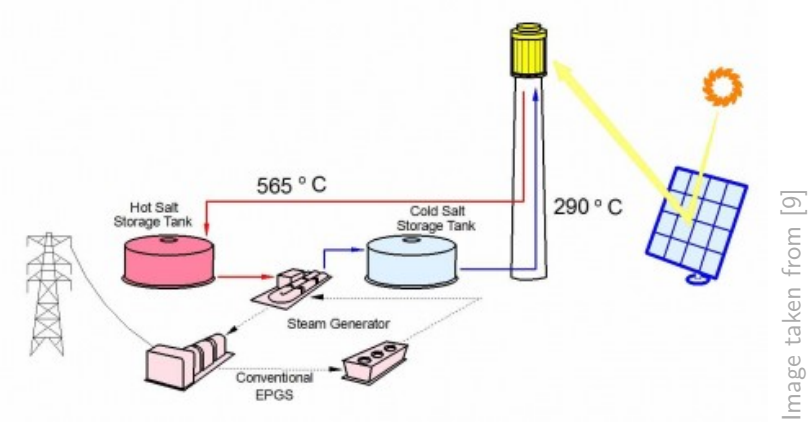


Figure 1: Schematic functioning of a solar thermal power plant with molten salts

The targeted objective of molten salt power plants is to take benefit of their high specific heat and hence to enable thermal energy storing. If energy storage doesn't increase the productivity of the plant (even worse, it deteriorates the global yield), it makes however possible a regulation of the energy production. That is, the output is no more entirely dependent on the incident radiation at a given moment of the day. This is especially profitable nowadays, where regulation of energy production has become a key issue for the energetic question. For example, molten salt solar power plants can manage a continuous generation of energy even during the night (when the source of energy has disappeared).

Unfortunately, the use of molten salts has serious negative impacts on heat

exchanger materials, which from now on have to operate in a highly corrosion environment. On top of that, heat exchangers are still subjected to diverse severe solicitations inherent to the functioning of solar power plants itself: cyclic loading, fatigue, creep, thermal gradients, etc. Note also that heat exchangers are not the single parts submitted to corrosion (e.g. steam generator which also perform at high temperature).

Hence, the integration of corrosion induced damage into the material behavior is even more important to consider that the aggressive potential of the working environment changes drastically.

We conclude this section mentioning that a such coupling model could find its use in various other applications involving corrosion and strength of materials. For example, in high pressure compressors and turbine design for civil aviation, higher levels of performance are reached by increasing the pressure ratios and higher turbine entry temperatures.

Procedure

During this study, the global problem has been split into several but simpler problems in order to facilitate the development. We have started with a one-dimensional model for corrosion diffusion, and extend it to two-dimensions. In parallel, the model for mechanical behavior (classical mechanics of solids) is developed. Once each model is validated, the coupling model is investigated. Each new model integrate thus an additional difficulty.

For each developed model, we tried to stick the following procedure

i **Development:** The equations are derived. Starting from an differential equation, the weak form of the problem is determined and subsequently the expressions of stiffness matrix and nodal load vector.

ii **Coding:** This stage is about pure numerical developments in Fortran and are not detailed in this report.

iii **Verification:** During this step, we will address the question to know if the equation are mathematically properly solved. Do we converge to a unique solution if the mesh is refined? Is the method sufficiently stable? Verification is hence about mathematics and practical and physical considerations will be meanwhile left apart[1].

iv **Validation:** The validation step adds the physical aspects of the phenomenon into consideration. These validation step is hence dedicated to determine whether the equation chosen are well adapted to model the considered physical phenomenon. For this purpose, numerical predictions will be compared with experimental results: if predictions match experimental data, the numerical model will be considered as validated.

[1] In the verification step, constants and physical quantities will be chosen to illustrate investigated phenomena and not necessarily to model the reality (for example, the reader should not be surprise to see that corrosion is fully developed in a 1 m long bar after several seconds)

Regarding the tools used for the coding stage, we used `gmsh` [10] for mesh generations. All calculations are carried on in `Fortran 90`, excepted for the post-processing stage, where a `Python` code has been developed. This code uses the `vtk` library for creating an unstructured grid exploitable by `Paraview` [11], which provides a convenient interface for data visualization.

Chapter 1

Transient model of corrosion diffusion in one dimension

1.1 Equation for corrosion diffusion

According to [2,3] the diffusion of corrosion is governed by the following equation

$$\frac{\partial c}{\partial t} = \nabla \cdot (D\nabla c - DMc\nabla p) \tag{1.1}$$

Where c represents a relative concentration in oxygen, p the isotropic pressure, D the diffusivity of the corrosion and M the pressure factor. In this first chapter, the only focus will be on the diffusion of the corrosion, neglecting all potential influences of the mechanical solicitations. Subsequently, a pressure factor of $M = 0$ will be assumed, in which case (1.1) is simplified to a purely diffusion equation (also called fickian diffusion) whose form is given by

$$\frac{\partial c}{\partial t} = \nabla \cdot (D\nabla c) \tag{1.2}$$

A constant diffusivity D will also be assumed everywhere inside the studied domain. In that case, the following equation will be obtained, which is analogous to the one describing transient heat conduction

$$\frac{\partial c}{\partial t} = D \cdot \nabla^2 c \tag{1.3}$$

Finally, the model studied in this section assumes the phenomenon to be unidimensional, that is

$$\frac{\partial c}{\partial t} = D \cdot \frac{\partial^2 c}{\partial x^2} \tag{1.4}$$

This last equation is a purely natural diffusive partial differential equation and highlights that the time derivative of the oxygen concentration is proportional to its Laplacian: the higher the average corrosion concentration around a given point, the faster the corrosion variation at this point. In other words, the value of c in one point will converge even faster to the steady state solution as the surroundings points are close to this solution.

1.2 Solving of the partial differential equation by finite elements method

1.2.1 Derivation of the weak form of the problem

The weak form of the problem can be obtained using the weighted residual method. We will search for an approximation function $\tilde{c}(x) \approx c(x)$. The weighted residual will be formed by considering this approximation function, multiplying equation (1.4) by one weighting function $\delta\tilde{c}(x)$, and integrating it over the studied volume V. These operations lead to the following equation

$$\texttt{WR} = \int_V D \cdot \frac{\partial^2 \tilde{c}}{\partial x^2} \cdot \delta\tilde{c}\, dV - \int_V \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c}\, dV \tag{1.5}$$

If the cross section A of the 1D studied body is constant along the length of the body, the integration over the volume results in the following expression

$$\Leftrightarrow \texttt{WR} = D \cdot \int_0^L \frac{\partial^2 \tilde{c}}{\partial x^2} \cdot \delta\tilde{c}\, dx - \int_0^L \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c}\, dx \tag{1.6}$$

The right hand side expression can be integrated by parts, which leads to

$$\texttt{WR} = D \cdot \left[\frac{\partial \tilde{c}}{\partial x} \cdot \delta\tilde{c}\right]_0^L - D \cdot \int_0^L \frac{\partial \tilde{c}}{\partial x} \cdot \frac{\partial \delta\tilde{c}}{\partial x}\, dx - \int_0^L \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c}\, dx \tag{1.7}$$

We finally impose the weighted residual WR to be equal to zero, to get

$$\texttt{WR} = 0 \quad \Leftrightarrow \quad D \cdot \left[\frac{\partial \tilde{c}}{\partial x} \cdot \delta\tilde{c}\right]_0^L - D \cdot \int_0^L \frac{\partial \tilde{c}}{\partial x} \cdot \frac{\partial \delta\tilde{c}}{\partial x}\, dx - \int_0^L \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c}\, dx = 0 \tag{1.8}$$

The last equation defines the weak form of the problem. The first term represents the effects of boundary conditions, and will be discussed more in detail in section 1.2.7. Meanwhile, it will be assumed to be zero. Thus, the weak form takes the following expression

$$- D \cdot \int_0^L \frac{\partial \tilde{c}}{\partial x} \cdot \frac{\partial \delta\tilde{c}}{\partial x}\, dx - \int_0^L \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c}\, dx = 0 \tag{1.9}$$

1.2.2 Derivation of the expressions of local stiffness matrix and local nodal load vector

Everywhere inside a given finite element, the unknown field $c(x)$ is approximated by a continuous field $\tilde{c}(x)$, expressed as a linear combination of the chosen shape functions:

$$c(x) \approx \tilde{c}(x) = \sum_{i=1}^{n_{dof}} N_i(x) \cdot q_i \tag{1.10}$$

Where $N_i(x)$ represents the shape function associated to the degree of freedom i, while q_i is the nodal unknown associated to the same degree of freedom. This

equation may be rewritten on a vectorial form that will be preferred for convenience in this document:

$$\tilde{c}(x) = \mathbf{N}\mathbf{q} \tag{1.11}$$

In this particular case with one single unknown by node, the matrix $\mathbf{N}$ is a row vector of n_{dof} components (size $1 \times n_{dof}$), while $\mathbf{q}$ is a column vector with the same number of components (size $n_{dof} \times 1$), where n_{dof} is the number of degree of freedom linked to an element. Therefore, we have:

$$\mathbf{N} = \begin{bmatrix} N_1(x) & N_2(x) & \cdots & N_{n_{dof}}(x) \end{bmatrix}, \qquad \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{n_{dof}} \end{bmatrix}$$

Before substituting the approximation function $\tilde{c}$ in the weak form, several additional quantities need to be expressed as a function on these $N_i(x)$. Similarly to (1.11), it is assumed that the weighting function $\delta\tilde{c}(x)$ can be expressed as

$$\delta\tilde{c}(x) = \sum_{i=1}^{n_{dof}} N_i(x) \cdot \delta q_i = \mathbf{N} \cdot \delta\mathbf{q} \tag{1.12}$$

From (1.11), it is readily proven that

$$\frac{\partial \tilde{c}}{\partial x} = \sum_{i=1}^{n_{dof}} \frac{\partial N_i}{\partial x} \cdot q_i \tag{1.13}$$

Once more, the vector notation will be preferred,

$$\frac{\partial \tilde{c}}{\partial x} = \mathbf{B}\mathbf{q} \qquad \text{where} \qquad B_i = \frac{\partial N_i}{\partial x} \tag{1.14}$$

And as previously, it is proven that

$$\frac{\partial \delta\tilde{c}}{\partial x} = \mathbf{B}\delta\mathbf{q} \tag{1.15}$$

Finally, note that because $c(x)$ and $\delta c(x)$ are scalar quantities, their transpos is equal to themselves, that is : $\delta c^T = \delta c$ and $c^T = c$ and subsequently : $\delta\mathbf{q}^T\mathbf{N}^T = \mathbf{N}\delta\mathbf{q}$ and $\mathbf{q}^T\mathbf{N}^T = \mathbf{N}\mathbf{q}$. Substituting all these previous quantities into (1.9), we get

$$-D\int_0^L \delta\mathbf{q}^T\mathbf{B}^T\mathbf{B}\mathbf{q}\,dx - \frac{\partial}{\partial t}\int_0^L \delta\mathbf{q}^T\mathbf{N}^T\mathbf{N}\mathbf{q}\,dx = 0 \tag{1.16}$$

$$\Leftrightarrow \delta\mathbf{q}^T\left[-D\int_0^L \mathbf{B}^T\mathbf{B}\mathbf{q}\,dx - \frac{\partial}{\partial t}\int_0^L \mathbf{N}^T\mathbf{N}\mathbf{q}\,dx\right] = 0 \tag{1.17}$$

The time derivative can be derived using first order backward finite difference:

$$\frac{\partial f}{\partial t} = \frac{f^t - f^{t-\Delta t}}{\Delta t} \tag{1.18}$$

Because $\mathbf{N}$ are only functions of x, the single variable affected by the time derivative are the nodal unknowns $\mathbf{q}$.

$$\Leftrightarrow \delta\mathbf{q}^T\left[-D\int_0^L \mathbf{B}^T\mathbf{B}\mathbf{q}^t\,dx - \int_0^L \mathbf{N}^T\mathbf{N}\left(\frac{\mathbf{q}^t - \mathbf{q}^{t-1}}{\Delta t}\right)\,dx\right] = 0 \tag{1.19}$$

Note that in this expression and the following, subscript t refers to the time t, while capital T denotes the transpose operation.

$$\Leftrightarrow \delta\mathbf{q}^T \left[\left(-\int_0^L D\Delta t\ \mathbf{B}^T\mathbf{B} + \mathbf{N}^T\mathbf{N}\ dx \right) \mathbf{q}^t + \left(\int_0^L \mathbf{N}^T\mathbf{N}\ dx \right) \mathbf{q}^{t-1} \right] = 0 \quad (1.20)$$

Noting that $\delta\mathbf{q}$ is arbitrary, the equation may be simplified as follow

$$\Leftrightarrow \left(\int_0^L D\Delta t\ \mathbf{B}^T\mathbf{B} + \mathbf{N}^T\mathbf{N}\ dx \right) \mathbf{q}^t = \left(\int_0^L \mathbf{N}^T\mathbf{N}\ dx \right) \mathbf{q}^{t-1} \quad (1.21)$$

This last equation highlights the expressions of stiffness matrix and nodal loads vector. The expression may then be rewritten as

$$\mathbf{K}_e\ \mathbf{q}_e^t = \mathbf{g}_e^t \quad (1.22)$$

With $\mathbf{K}_e$ the elementary stiffness matrix and $\mathbf{g}_e$ the elementary consistent nodal loads vector whose expressions are given by

$$\mathbf{K}_e = \int_0^L D\Delta t\ \mathbf{B}^T\mathbf{B} + \mathbf{N}^T\mathbf{N}\ dx \quad (1.23)$$

$$\mathbf{g}_e^t = \int_0^L \mathbf{N}^T\mathbf{N}\mathbf{q}^{t-1}\ dx \quad (1.24)$$

Equation (1.22) shows that the approximation solution can be found by solving a linear system where the independent term changes at each iteration (because dependent of the solution $\mathbf{q}$ at previous step).

1.2.3 Stability requirements

This section is dedicated to the study of the integration scheme described previously.

By inspection, equation (1.17) may be written on the following very general form,

$$\mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g} \quad (1.25)$$

We assumed that the time derivative was discretized according the a backward-euler finite differences,

$$\mathbf{C}\left(\frac{\mathbf{q}^{t+1} - \mathbf{q}}{\Delta t} \right) + \mathbf{K}\mathbf{q} = \mathbf{g} \quad (1.26)$$

However, the discretization scheme gives no information on the time at which should be calculated $\mathbf{K}\mathbf{q}$. In fact two, choices are possible

$$\begin{cases} \mathbf{C}\mathbf{q}^{t+1} = \mathbf{C}\mathbf{q}^t - \Delta t\mathbf{K}^t\mathbf{q}^t + \mathbf{g}^t & (1.27a) \\ \mathbf{C}\mathbf{q}^{t+1} = \mathbf{C}\mathbf{q}^t - \Delta t\mathbf{K}^{t+1}\mathbf{q}^{t+1} + \mathbf{g}^{t+1} & (1.27b) \end{cases}$$

The first one is called an explicit integration scheme, the nodal unknown at time $\mathbf{q}^{t+1}$ depends only on the nodal unknown at previous time $\mathbf{q}^t$. One of its main advantage is that the iteration can be evaluated based on a iteration matrix. Provided that the stiffness matrix $\mathbf{K}$ is constant over time, explicit discretization

schemes require to determine an iteration matrix once by solving a linear system, and after, the nodal unknown at time $t+1$ are simply obtained from a matrix multiplication. This explicit discretization scheme is however assorted to stability conditions [12], stating that the chosen Δt should not exceed the value of a critical time step. The second option is an implicit integration scheme. This one is unconditionally stable, but requires a higher computational cost. The discretization schemes chosen for all models of this work will be implicit.

1.2.4 Derivation of the expressions of global stiffness matrix and global load vector

Now that the expressions of local stiffness matrix and local nodal consistent load vector are derived, the studied domain can be extended to a mesh of finite elements: that is, the studied domain can be discretized in finite elements whose stiffness matrices and nodal loads vectors are known by (1.23) and (1.24). To build the global stiffness matrix knowing each local stiffness matrix $\mathbf{K}_e$, it is necessary to perform an assembling operation over each $\mathbf{K}_e$. The same operation is executed to get $\mathbf{g}^t$

$$\mathbf{K} = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \mathbf{K}_e = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \left(\int_0^L D\Delta t \, \mathbf{B}_e^T \mathbf{B_e} + \mathbf{N}_e^T \mathbf{N}_e \, dx \right) \tag{1.28}$$

$$\mathbf{g}^t = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \mathbf{g}_e = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \left(\int_0^L \mathbf{N}_e^T \mathbf{N}_e \mathbf{q}_e^{t-1} \, dx \right) \tag{1.29}$$

1.2.5 Shape functions

Because the weak form of the problem (1.9) involves first order spatial derivative, the shape functions used to build the approximation must be at least linear to build a sufficiently rich polynomial approximation of $c(x)$. A generic example of linear shape functions associated to element i are illustrated in Figure 1.1, for a one-dimensional structured mesh. Their mathematical expressions are given by

$$N_1(x) = \frac{x}{l} \qquad\qquad N_2(x) = 1 - \frac{x}{l}$$

With $x \in [0,\ l]$. Alternatively, the shape functions can be derived as functions of an adimensional variable $\xi \in [-1,\ 1]$ in order to facilitate the numerical integration that will be performed in the further steps. Assuming $\xi = \dfrac{2x}{L} - 1$, we get

$$N_1(\xi) = \frac{1}{2}(1 - \xi) \qquad\qquad N_2(\xi) = \frac{1}{2}(1 + \xi)$$

Using the adimensional coordinate system, the first derivative of the shape function matrix $\mathbf{N}$ can be derived using chain rule derivative,

$$B_i(\xi) = \frac{\partial N_i(\xi)}{\partial x} = \frac{\partial N_i}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} \tag{1.30}$$

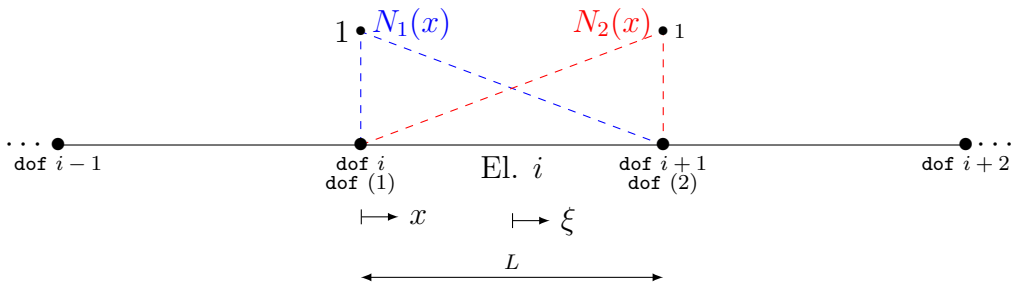


Figure 1.1: Linear shape functions associated to element i (between brackets, local node numbering relative to an element)

After calculation of first derivatives, we will obtain the expressions of the matrices $\mathbf{N}$ and $\mathbf{B}$

$$\text{for } x \in [0,\ L]: \qquad \mathbf{N} = \begin{bmatrix} \frac{x}{L} & 1 - \frac{x}{L} \end{bmatrix} \qquad \mathbf{B} = \frac{1}{L}\begin{bmatrix} -1 & 1 \end{bmatrix} \tag{1.31}$$

$$\text{for } \xi \in [-1,\ 1]: \qquad \mathbf{N} = \frac{1}{2} \cdot \begin{bmatrix} 1 - \xi & 1 + \xi \end{bmatrix} \qquad \mathbf{B} = \frac{1}{L}\begin{bmatrix} -1 & 1 \end{bmatrix} \tag{1.32}$$

Using adimensional coordinate system with variable ξ, the expression of the local stiffness matrix $\mathbf{K}_e$ (1.28) can be transformed into

$$\mathbf{K}_e = \int_{-1}^{1} \left(D\Delta t\ \mathbf{B}^T\mathbf{B} + \mathbf{N}^T\mathbf{N} \right) \cdot J\ d\xi \tag{1.33}$$

Where the jacobian J of the regular transformation can be determined noting that

$$\xi = \frac{2x}{L} - 1 \quad \longrightarrow \quad d\xi = \frac{2}{L}\, dx$$

And then, we get the final expression of the local stiffness matrix

$$\mathbf{K}_e = \frac{L}{2} \int_{-1}^{1} \left(D\Delta t\ \mathbf{B}^T\mathbf{B} + \mathbf{N}^T\mathbf{N} \right)\ d\xi \tag{1.34}$$

Moreover, using the same change of variable than in (1.29),

$$\mathbf{g}_e^t = \frac{L}{2} \int_{-1}^{1} \mathbf{N}^T\mathbf{N}\mathbf{q}^{t-1}\ d\xi \tag{1.35}$$

1.2.6 Numerical Integration

The evaluation of local stiffness matrices and nodal load vectors through equations (2.37) and (2.38) require both the calculation of an integral. For this purpose, the Gauss quadrature (also known as Gaussian-Legendre quadrature if the integral bounds are $[-1,\ 1]$) is chosen. This method is more efficient than the "classical" trap method because of its ability to integrate exactly polynomial functions, provided that a sufficient number of Gauss points are chosen, and benefits a higher accuracy for fewer functions calls.

Figure 1.2 compares the integration of a given 3$^{\text{rd}}$ order polynomial by both methods. It shows that an optimal choice of evaluation points, as suggested by

the Gauss quadrature method (rather than equally spaced points), improves the results of the calculation. Further information concerning numerical integration may be found in [13, 14].

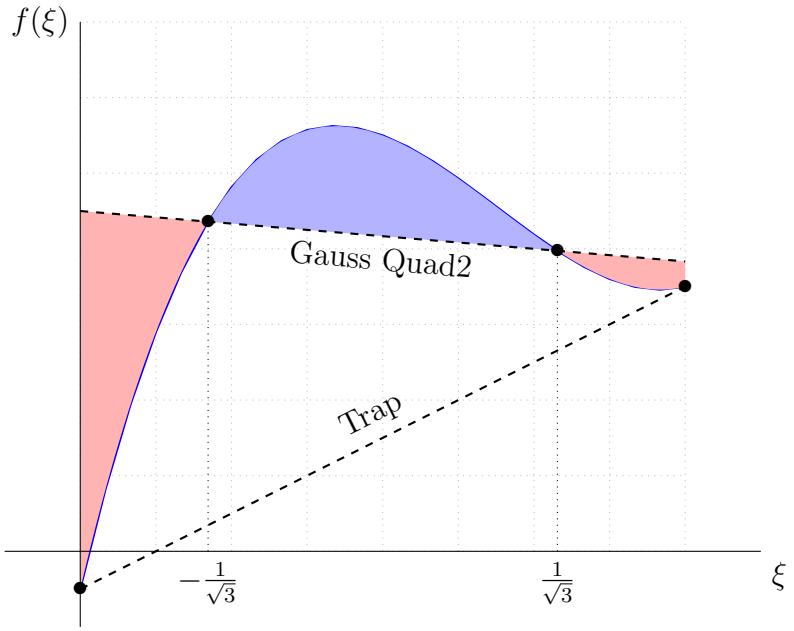


Figure 1.2: Gauss-Legendre quadrature with 2 Gauss points vs. trap method illustrated on function $f(\xi) = 7\xi^3 - 8\xi^2 - 3\xi + 3$ with $\xi \in [-1,\ 1]$.

The Gauss quadrature method states that a single variable function $f(\xi)$ can be integrated as follow

$$I(\eta) = \int_{-1}^{1} f(\xi)\, d\xi \approx \sum_{i=1}^{n} H_i^* \cdot f(\xi_i^*) \tag{1.36}$$

Where ξ_i^* (with $i = 1, \ldots, n$) are n abscissa defining the location of the n Gauss points. These abscissa and associated weights are related to the Legendre polynomials: the abscissa $\xi_1^*, \xi_2^*, \ldots, \xi_n^*$ are the roots of the nth Legendre polynomial $P_n(\xi)$ while the associated weights H_i^* are defined by

$$H_i^* = \int_{-1}^{1} \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{t - \xi_i}{\xi_i - \xi_j}\, dt \tag{1.37}$$

The calculation of these roots and weights for low order Legendre polynomials offers nowadays little interest as numerous tables already extensively set them out. Hence, Table 1.1, adapted from [14] will be used in this work.

Regarding the precision of the integration, it is known by [13, 15] that if n refers to the number of Gauss points used, the integration of a single variable polynomial of degree of at most $p = 2n - 1$ is exactly performed. In the present case, the highest order terms of the polynomial $f(\xi)$ originate from the product $\mathbf{N}^T\mathbf{N}$ appearing in both (1.34) and (1.35). Because the shape function matrix $\mathbf{N}$ contains linear polynomial, the highest order of $f(\xi)$ is quadratic in ξ. Hence, the highest order is given by $p = 2$ and a minimum of 2 Gauss points ($n = 2$) are required to perform an exact calculation of the integral following equation (2.39).

n	Roots ξ_i^*	Weight H_i^*
2	0.5773502692	1.0000000000
	-0.5773502692	1.0000000000
3	0.7745966692	0.5555555556
	0.0000000000	0.8888888889
	-0.7745966692	0.5555555556
4	0.8611363116	0.3478548451
	0.3399810436	0.6521451549
	-0.3399810436	0.6521451549
	-0.8611363116	0.3478548451
5	0.9061798459	0.2369268850
	0.5384693101	0.4786286705
	0.0000000000	0.5688888889
	-0.5384693101	0.4786286705
	-0.9061798459	0.2369268850

Table 1.1: Roots and weights of nth Legendre polynomials for $n = 2, 3, 4$ and 5.

1.2.7 Boundary Conditions

Reminding that (1.9) has been obtained neglecting the contribution of boundary conditions, it is now necessary to formalize the reasons of this choice. The strong form governs the behavior of the body in any point of an infinite one dimensional domain. Since the studied domain is finite, this equation can no longer be representative for the extremities of the bar. This shortcoming is made up by the weak form statement (1.8) which accounts for the exchanges with the exterior domain trough the boundaries. In the present applications, exchanges with the exterior are modeled by a corrosion flux[1] $\dfrac{\partial c}{\partial x}$.

If the situation is such that there is no exchange of energy with the exterior (for example if a half part of an axi-symmetrical body, with axi-symmetrical loading are studied), the flux will be zero. Otherwise, the exchange is modeled by a non zero $D\left[\dfrac{\partial \tilde{c}}{\partial x} \cdot \delta\tilde{c}\right]$. The value of $\dfrac{\partial c}{\partial x}$ is an input of the code, specified by user who model the influence of the external world at appropriate (and chosen) locations.

Mathematically, the influence of boundary condition may be summarized as follow

$$D\frac{\partial \tilde{c}}{\partial x} \cdot \delta\tilde{c} = D\delta\mathbf{q}^T\mathbf{N}^T\frac{\partial \tilde{c}}{\partial x} \tag{1.38}$$

Substituted back in the weak form statement (1.8), and performing the same operations as the ones done previously on (1.9) leads to

$$\mathbf{K}_e\mathbf{q}_e = \mathbf{g}_e^t + D\Delta t\boldsymbol{\phi}_c \tag{1.39}$$

[1] A corrosion flux is strictly speaking meaningless, at least for physical considerations: it is certainly more suitable to speak about a oxygen flux. But since the fundamental variable chosen for corrosion diffusion is the corrosion rate c itself, the denomination of corrosion flux will be kept, bearing in mind that both fluxes are strongly related.

With $\boldsymbol{\phi}_{ci}$ is a row vector containing the fluxes $\dfrac{\partial \tilde{c}}{\partial x}$ injected in node i from outside of the domain. This vector is empty except for at most two components (those related with the nodes located at both extremities). Fluxes are then considered as loads and simply added to the nodal load vector.

1.3 Model verification

This section is dedicated to the study of the numerical results provided by the implemented code. This verification focuses only on mathematical aspects and **will not consider the physical aspects of the problem**. The aim of this section is to address the ability of the code to solve the equation (1.4) properly.

1.3.1 Reference solutions

In order to assess the quality of the solution provided by the model detailled in the previous section, the present solution will be compared with a reference solution. Since the studied problem is uni-dimensional, it is still reasonable to search for an analytical solution. The analytic development explained hereafter depends on the type of boundary conditions enforced: in order to assess the ability of the developed model to accommodate both essential and natural boundary conditions, two test cases will be considered and confronted with their respective analytical solution. For this reason, the analytical developments will be split into two sections.

First solution (essential BC's)

We are searching for a the function $c(x,t)$ such that

$$\frac{\partial c}{\partial t} = D \cdot \frac{\partial^2 c}{\partial x^2} \tag{1.40}$$

With initial conditions given by a general function of x and constant boundary conditions

$$\begin{cases} c(0,t) = c_{x_0} & \text{(1.41a)} \\ c(L,t) = c_{x_L} & \text{(1.41b)} \\ c(x,0) = f(x) & \text{(1.41c)} \end{cases}$$

The non-homogeneous initial and boundary conditions add a certain complexity to this problem. To make it simpler, it is suggested to split it into two independent and simpler problems. Firstly a non-homogeneous steady state problem, whose solution is given by function $d(x)$and with boundary conditions:

$$d(0) = c_{x_0} \qquad d(L) = c_{c_L} \tag{1.42}$$

The second problem is a homogeneous transient problem whose solution is described by function $c^*(x,t)$ with initial conditions

$$c^*(x,0) = F(x) = f(x) - d(x) \qquad c^*(0,t) = 0 \qquad c^*(L,t) = 0 \tag{1.43}$$

The two previous problems are coupled by the following equation, expressing that the solution of problem (1.40) is the sum of a spatial and time-dependent function and a purely spatial function.

$$c(x,t) = d(x) + c^*(x,t) \tag{1.44}$$

Substituting $c(x,t)$ into (1.40), we get

$$\frac{\partial d}{\partial t} + \frac{\partial c^*}{\partial t} = D \cdot \left(\frac{\partial^2 d}{\partial x^2} + \frac{\partial^2 c^*}{\partial x^2} \right)$$

This equation is split into a system of two equations, each of them addressing one of the two problems described previously

$$\begin{cases} \dfrac{\partial^2 d}{\partial x^2} = 0 & \text{(1.45a)} \\ \dfrac{\partial c^*}{\partial t} = D \cdot \dfrac{\partial^2 c^*}{\partial x^2} & \text{(1.45b)} \end{cases}$$

Equation (1.45a) is readily solved by,

$$d(x) = Ax + B$$

Enforcing the boundary conditions (1.42),

$$d(x) = (c_{c_L} - c_{x_0}) \cdot \frac{x}{L} + C_{x_0} \tag{1.46}$$

The time-dependent function (1.45b) can now be solved. Assuming the separation of variable

$$c^*(x,t) = \phi\,(x) \cdot \Gamma(t) \tag{1.47}$$

where the concentration is a function of x whose amplitude is modulated by $\Gamma(t)$, a time-dependent function (principle analogous to the one followed in the theory of vibration of continuous systems [16]). Substituting (1.47) into (1.45b), we get

$$D\Gamma(t) \cdot \frac{\partial^2 \phi}{\partial x^2} = \phi(x) \cdot \frac{\partial \Gamma}{\partial t} \Leftrightarrow \frac{D}{\phi(x)} \cdot \frac{\partial^2 \phi}{\partial x^2} = \frac{1}{\Gamma(t)} \cdot \frac{\partial \Gamma}{\partial t} \tag{1.48}$$

As ϕ is a function of x only, the left hand side of (1.48) is only a function x, while the right hand side must be a function of t. Given that (1.48) imposes both terms to be equal, it results that they are both constant. Assuming this constant to be $-\omega^2$,

$$\frac{D}{\phi(x)} \cdot \frac{\partial^2 \phi}{\partial x^2} = \frac{\partial \Gamma}{\partial t} \cdot \frac{1}{\Gamma(t)} = -\omega^2 \tag{1.49}$$

Last equation can be one transformed into te following system,

$$\begin{cases} D\dfrac{\partial^2 \phi}{\partial x^2} + \omega^2 \phi(x) = 0 & \text{(1.50a)} \\ \dfrac{\partial \Gamma}{\partial t} + \omega^2 \Gamma(t) = 0 & \text{(1.50b)} \end{cases}$$

Both equations can be solved independently using classical method for linear differential equations, to get

$$\begin{cases} \phi(x) = A^* \cdot \sin(k\omega x) + B^* \cos(k\omega x) & \text{(1.51a)} \\ \Gamma(t) = C^* \cdot \exp\left(-\omega^2 t\right) & \text{(1.51b)} \end{cases}$$

with $k = 1/\sqrt{D}$. Substituting these solutions in (1.47), one get

$$c^*(x,t) = [A \cdot \sin(k\omega x) + B \cdot \cos(k\omega x)] \cdot \exp\left(-\omega^2 t\right) \tag{1.52}$$

After, the boundary conditions (1.43) are imposed: from $c^*(0,t) = 0$, we get $B = 0$. The second boundary conditions $c^*(L,t) = 0$ gives[2]

$$A \cdot \sin(k\omega L) = 0 \quad \Rightarrow \quad k\omega L = n\pi \quad (n = 0, 1, \dots) \tag{1.53}$$

Hence, at each value of n corresponds one solution of the boundary conditions. As a result, it exists an infinite number of solution $c_n(x)$ of (1.40):

$$c_n(x,t) = A_n \cdot \sin\left(\omega_n k x\right) \cdot \exp\left(-\omega_n^2 t\right) \tag{1.54}$$

With

$$\omega_n = \frac{n\pi}{kL} \qquad (n = 0, 1, \dots) \tag{1.55}$$

The superposition of every $c_n(x)$ gives

$$c^*(x,t) = \sum_{n=1}^{\infty} c_n^*(x,t) = \sum_{n=1}^{\infty} A_n \sin\left(\frac{n\pi x}{L}\right) \cdot \exp\left(-\frac{n^2\pi^2}{k^2 L^2} t\right) \tag{1.56}$$

The constants A_n are then determined imposing the initial conditions,

$$c^*(x,0) = F(x) = \sum_{n=1}^{\infty} A_n \cdot \sin\left(\frac{n\pi x}{L}\right) \tag{1.57}$$

This last equation is by definition the Fourier sine series of function $F(x)$ and A_n is therefore expressed by

$$A_n = \frac{2}{L} \int_0^L F(x) \cdot \sin\left(\frac{\pi n x}{L}\right) \, dx \tag{1.58}$$

Finally, substituting into (1.44) gives the final solution

$$c(x,t) = c_{x_0} + (c_{x_L} - c_{x_0}) \cdot \frac{x}{L} + \sum_{n=1}^{\infty} A_n \cdot \sin\left(\frac{n\pi x}{L}\right) \cdot \exp\left(-\frac{n^2\pi^2}{k^2 L^2} t\right) \tag{1.59}$$

Second solution (natural BC+essential BC)

A second study case will be considered with one boundary condition of type "flux", i.e. such that

$$\begin{cases} \dfrac{\partial c}{\partial x}(0,t) = \phi_0 & \text{(1.60a)} \\ c(L,t) = c_{x_L} & \text{(1.60b)} \\ c(x,0) = f(x) & \text{(1.60c)} \end{cases}$$

In that case, the solution is derived in an analogous manner, and takes the final form,

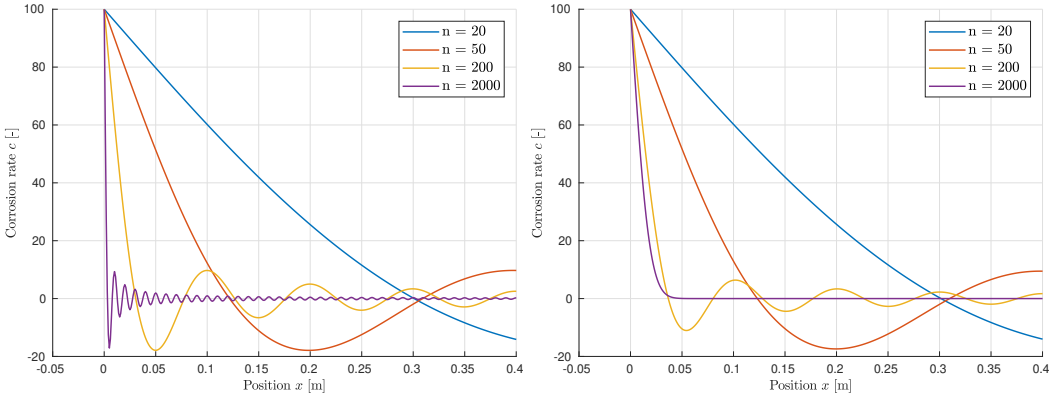
$$c(x,t) = c_{x_L} + \phi_0 \left(x - L\right) + \sum_{n=1}^{\infty} A_n \cdot \cos\left[\left(n - \frac{1}{2}\right)\frac{\pi x}{L}\right] \cdot \exp\left[-\left(n - \frac{1}{2}\right)^2 \frac{\pi^2}{k^2 L^2} t\right] \tag{1.61}$$

[2]The solution $A = 0$ is set apart because trivial and non interesting.

With

$$A_n = \frac{2}{L}\int_0^L F(x)\cdot\cos\left[\left(n-\frac{1}{2}\right)\frac{\pi x}{L}\right]\,dx \tag{1.62}$$

Note that both equations (1.59) and (1.61) include a Fourier series. Because it is numerically impossible to deal with a semi infinite sum, a truncation of these sums must be operated. In that case, a deterioration of the analytical solution will appear, especially in zone of discontinuities or brutal changes of $c(x)$ (describing the initial conditions). This phenomenon, well known when working with Fourier series is called Gibbs phenomenon and causes larger deterioration of the solution as the truncation in n increases and as the variation of the function is steep. Subsequently, it must be borne in mind that the analytical solution can be less reliable in zones where the corrosion level c is expected to change rapidly. This phenomenon is illustrated in Figure 1.3 for two different time values. As it can be seen, its effects become rapidly negligible once the discontinuities in c have been softened (in the situation illustrated, after 0.001 seconds, the Gibbs phenomenon is no more detectable for $n = 2000$).



(a) At $t = 10^{-6}$ s (b) At $t = 10^{-3}$ s

Figure 1.3: Illustration of Gibbs phenomenon at different time values, and for different truncation of the Fourier series (used: $c_{x_0} = 100\%$, $c_{x_L} = 25\%$ and $c_0 = 0\%$ and $D = 0.1\,\mathrm{m^2\,s^{-1}}$)

1.3.2 Study case 1

The first study case consists in a one meter long bar, subjected to two essential boundary conditions. The situation is schematically shown in Figure 1.4. The

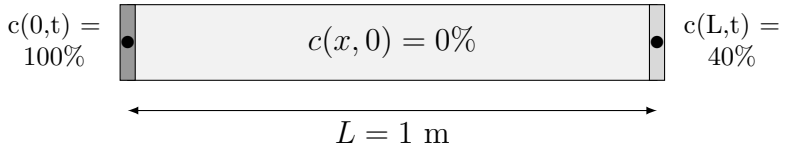
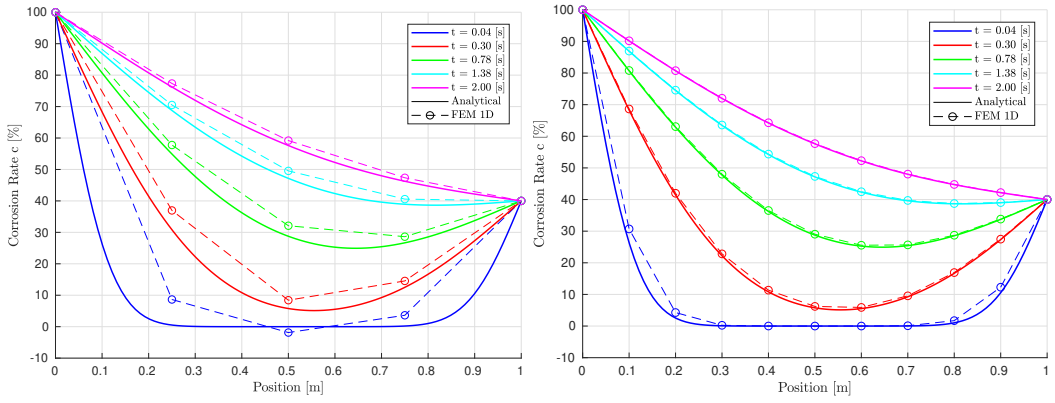
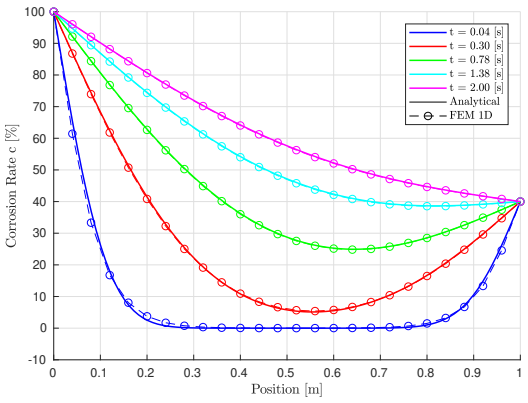


Figure 1.4: Initial and boundary conditions for study case 1

results are directly obtained from `Fortran` code, before being submitted to post processing in Matlab. They are displayed in Figure 1.5 and compared to the analytical solution (1.61), for 4, 10 and 25 elements.



(a) With 4 elements (b) With 10 elements



(c) With 25 elements

Figure 1.5: Comparison between analytical and numerical solutions for study case 1 ($\Delta t = 0.02$, $D = 0.1\ \mathrm{m^2\,s^{-1}}$)

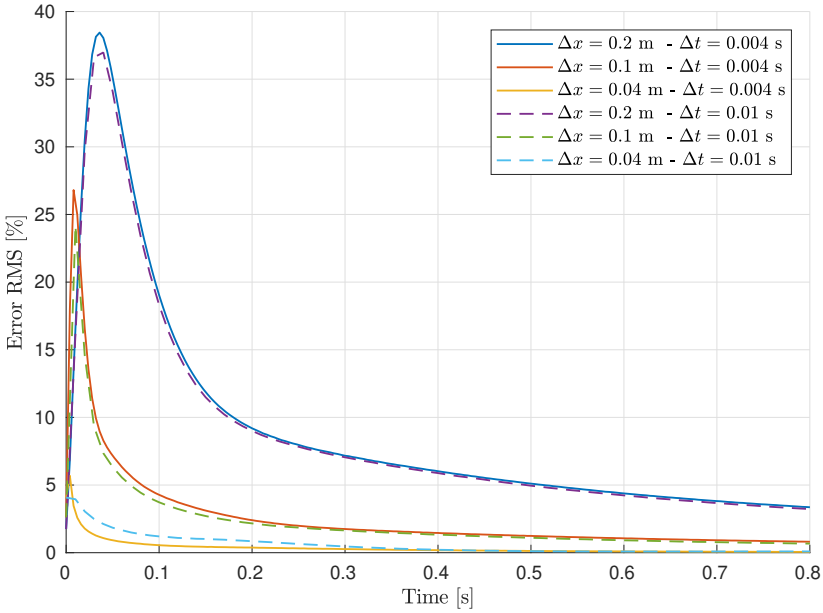


Figure 1.6: Evolution of the error with the time for first test case with $t \in [0,\ 1]$ (Parameters used in this Figure have been adapted because error definition is not defined if $c_0 = 0$. The initial and boundary conditions have all been slightly shifted by $c = 10$. Thus parameters used for this figure are: $c_1 = 110\%$, $c_2 = 50\%$, $c_0 = 10\%$, $D = 0.1\ \mathrm{m^2\,s^{-1}}$)

Some observations can be made when comparing these 3 figures.

- The higher the number of elements, the closer the results of the FEM method to the analytical solution. This is of course expected, as a finer mesh must provide a more accurate solution.

- As the time increases, the transient term of (1.4) becomes more and more negligible and one progressively tends to the steady state solution. For such boundary conditions, this solution is linear.

- For a small number of elements (e.g. Figure 1.5 (a)), the results are less accurate for small values of t (i.e. the first iterations). In these time regions, the actual state of the system (=initial state) is far from the steady state regime, and the evolution is governed by the transient term. Mathematically, the quantity $\dfrac{\partial^2 c}{\partial x^2}$ in equation (1.40) generates a large time variation in corrosion rate to satisfy the equality. There is an analogy with mechanical systems: a large force will generate a large acceleration of the mass. The use of a small Δt would therefore be suggested to model properly the motion in time zones where rapid changes are expected.

  However, this is not the case as the integration scheme is not responsible for these errors. For further explanations, Figure 1.6 will be considered, where an evolution of the error is displayed as a function of time. The error chosen to characterize the quality of the numerical solution is

$$\texttt{Error}(t) = \sqrt{\frac{1}{N_{dofs}} \sum_{i=1}^{N_{dofs}} \left(\frac{c_{th}(t) - c_{FEM}(t)}{c_{th}(t)} \right)^2} \tag{1.63}$$

  The 3 solutions (for 5, 10 and 25 elements) experience all a large error for small values of time, indicating incontestably that the FEM solution is less reliable in this time regions. Comparing the different curves shown in Figure 1.6 shows that opting for a smaller Δt does not change the amplitudes of the peaks[3]. The errors are however considerably minimized increasing the number of elements in the mesh. It results that the most efficient way to minimize the error is to refine the spatial mesh, rather than the time step.

  Notice that, since we are trying to estimate an error for small values of time, a special attention has been payed to the n number of coefficients in the Fourier series during the computation of Figure 1.6 (cfr. (1.59) and (1.61)). A value of $n = 300$ has been chosen, higher values leading to unnecessary calculations in the post-processing. Figure A.1 at page A1 illustrates the Gibbs phenomenon for several values of n at $t = 4 \times 10^{-3}$ (i.e. the first time value that we considered in Figure 1.6) and prove that $n = 300$ gives entire satisfaction.

- The differences in amplitudes of the peaks show how important adopting a fine mesh to approach the analytical solution is, especially in zones of high

[3] A direct comparison of the amplitudes of the peaks may be confusing at the first glance, as the time step is coarser for the dotted lines and the error is represented with a lower resolution. This low resolution averages the amplitude of the real peak: some differences are hence observed in amplitudes with curves related to $\Delta t = 0.004$ but we can not guarantee that they are "real"

corrosion gradient. Remember that the finite element method imposes the weak form equation to be satisfied **on average** inside an element: if a coarse mesh is chosen, the approximate solution will inadequately represent zones of high corrosion gradient. As a results: the error will always be negatively affected by a peak whose amplitude can be bounded by an adequate choice of Δx.

- Another interesting feature to note from Figure 1.5, is that the error committed during the first iterations do not compromise the quality of the approximated solution further in the iterations: the higher the time the closer to the analytical solution. The error is then not accumulated with the time. This can also be observed in Figure 1.6, with the horizontal asymptotic behavior for t tending to infinity.

1.3.3 Study case 2

The first study case examined a situation with only essential boundary conditions. Here, we will a study another situation mixing both essential and natural boundary conditions. As a reminder, essential boundary conditions are conditions applied on unknown field (so one value of c is imposed at one boundary) while natural boundary conditions are applied on its first spatial derivative (flux boundary condition). The scheme represented at Figure 1.7 represents the considered case. A flux of 2.5m^{-1} and a constant corrosion rate $c = 25\%$ are applied respectively at the left and right boundaries of the one dimensional bar.

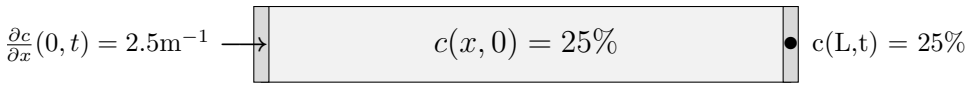
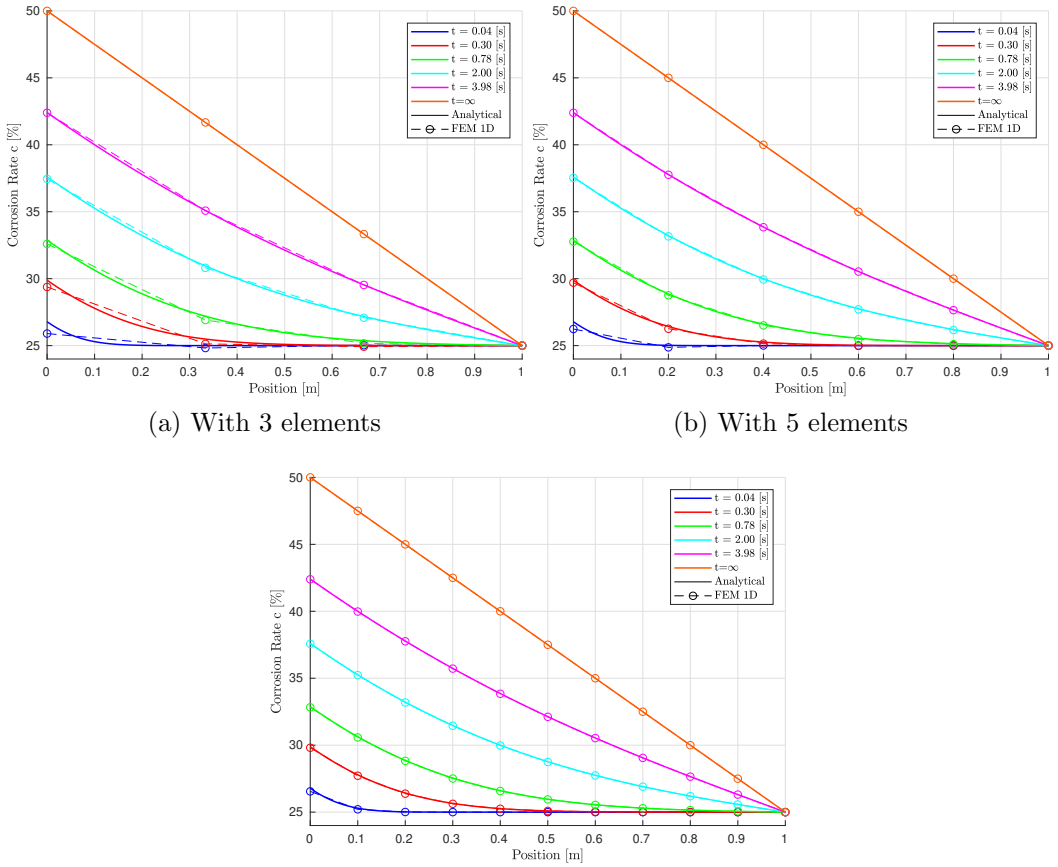


Figure 1.7: Initial and boundary conditions for study case 2

Results obtained from the code are shown in Figure 1.8. For 5 elements meshes or more, it provides nearly superimposed curves so that it is nearly impossible to distinguish theoretical from numerical curve especially for times far from the initial time. Once more, the numerical solution is less accurate in zones where the corrosion gradient appears to be high. The closer the corrosion rate from the steady state solution, the smaller the gradient and the smaller the error. Thus, here again, the code accommodates better gradients of corrosion when a finer mesh is chosen. Figure 1.9 displays the time evolution of the RMS global error, for 3, 5 and 10 elements meshes. The error is larger for the 3 nodes meshes as expected. The graph is also marked by peaks similar to the ones observed in Figure 1.6: their presence must be attributed to the same reasons mentioned earlier.

The presence of a flux at the left boundary increases gradually the corrosion rate c. This increase in c propagates trough the whole domain excepted at the right boundary where a corrosion rate is imposed constant at 25%. Corrosion increases up to a certain point, when the steady state solution is reached: the final distribution of c is linear across the domain (the slope is given by ϕ/D with ϕ the global flux resulting from the essential and natural boundary conditions applied on both extremities).



(a) With 3 elements

(b) With 5 elements

(c) With 10 elements

Figure 1.8: Comparison between analytical and numerical solutions for study case 2 (parameters: $\Delta t = 0.02$ s, $D = 0.1$ m^2 s^{-1}, $c(L,t) = 25\%$, $c(x,0) = 25\%$ and flux $\phi_c(0,t) = 2.5$ m^{-1})

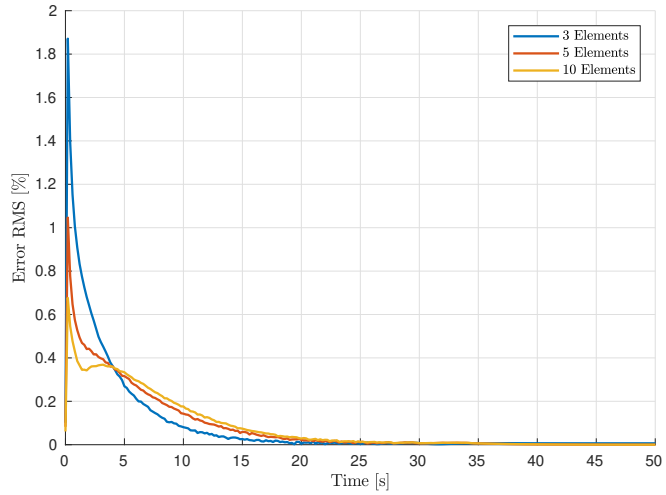


Figure 1.9: Time evolution of global RMS error for study case 2(parameters: $\Delta t = 0.2$ s, $D = 0.1$ m^2 s^{-1}, $c(L,t) = 25\%$, $c(x,0) = 25\%$ and flux $\phi_c(0,t) = 2.5$ m^{-1})

Note finally that several easy but fast checks can be done on the aspect (and also values) of both numerical results and analytical solution (at least regarding the steady state solution), by using an analogy with heat transfer equation. The

fundamental equation of heat diffusion, which is identical to (1.4), is:

$$k \cdot \frac{\partial^2 T}{\partial x^2} = \frac{\partial T}{\partial t} \tag{1.64}$$

In the particular case of a steady state regime, the Fourier law is recovered by integration, which establishes [17] a link between the gradient of temperature (by analogy with the gradient of corrosion) and the heat flux (by analogy with the corrosion flux)

$$q_x = k\frac{dT}{dx} \Rightarrow \Delta T = \frac{q_x}{k} \cdot \Delta x \tag{1.65}$$

Applied to our study case, we obtain in steady state:

$$\phi_c = D \cdot \frac{\partial c}{\partial x} \Rightarrow \Delta c = \frac{\phi_c}{D}\Delta x = \frac{2.5}{0.1} = 25 \tag{1.66}$$

Which is indeed the difference in c observed in Figure 1.8. This analogy can be useful to perform fast physical checks of the obtained results (example: steady state linear distribution as in Figure 1.8, etc, ...) and is of course valid whatever the study case, because it is inherent to the natural diffusion phenomenon.

1.4 Model validation

In this section, numerical results provided by the implemented code will be compared with experimental results: whilst the question addressed in section 1.3 was to determine whether the code was correctly implemented (i.e. the mathematical problem was correctly solved), this section focuses on determining if the natural diffusion process does correctly match with the corrosion diffusion phenomenon.

Due to the short duration of this master thesis, it was not envisaged to carry out experiments. Therefore experimental data have been obtained from literature where several references used to study hot corrosion of superalloys.

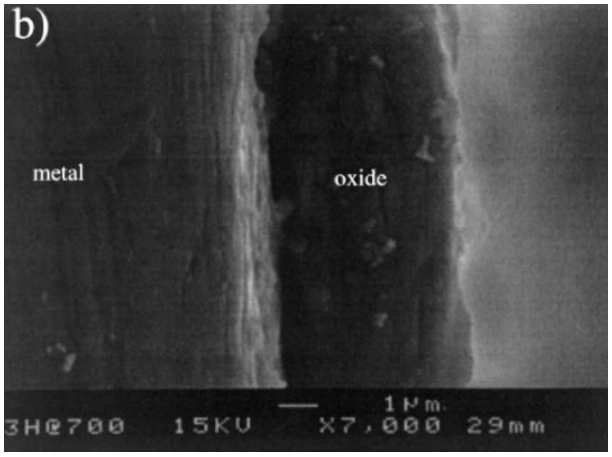
1.4.1 Few comments about experimental data

All the reviewed experiments consist of samples placed in a furnace at various constant temperatures for different periods of time. Therefore, corrosion takes place uniformly from all the boundaries of the sample and diffuses directly into the core, which consists in one-dimensional corrosion diffusion process. Such studies may then only be used to validate the 1D model, as they involve only 1D diffusion.

In time, corrosion progress is usually quantified by measuring the evolution of a weight gain [18] and a descaling loss. The descaling weight[4] (or weight loss) is the mass per unit ($\mathrm{mg\,cm^{-2}}$) of area lost during the descaling operation which consists in the removal of all corrosion products present on the sample surface (such as sulfides, oxides, ...). Thus, this descaling operation does not affect the precipitates present in the metal matrix (internal oxides), but only oxides

[4]The term "weight" is commonly employed in the literature, even if improper.

concerned by spallations. Figure 1.10 shows a photograph of the oxide layer and of the internal oxides present in the metal matrix of a a TI-15-3 alloy, after 3 hours at 700 °C. The weight gain (mg cm^{-2}) refers to the change in mass due to



Photograph taken from [3]

Figure 1.10: S.E.M. photograph of the oxide layer thickness, with internal oxides formed after 3h in 700 °C

oxidation and is mathematically expressed as

$$\Delta m = \frac{m_f - m_i}{S_0} \tag{1.67}$$

With m_f and m_i respectively the final and initial mass of the sample, and S_0 the initial sample surface[5]. This weight gain is hence a measure of the mass gain due to internal corrosion. Such approach can not directly be used, as our numerical model predicts corrosion rate evolution: it provides thus information regarding the oxygen concentration rather than about the mass of oxides formed during the corrosion process. However, a relationship between corrosion rate and oxides mass can be found [1], accounting for the density of the formed oxides.

Alternatively, the depth of corrosion can be determined by *focused ion beam* (FIB) measuring the thickness of oxide layer or even by spectroscopy giving the EDS map. These two methods provide a local information for both the internal oxides depth and oxidation layer thickness.

In order to proceed to a relevant the comparison with experimental data, two last issues need to be discussed. The first is about the type of boundary conditions. Several authors [19], [3] and [20] have opted for specifying c at the boundary (essential boundary conditions) while [2] described that the corrosion process is instigated by a time-dependent flux imposed at the boundary. The expression of this flux is readily determined from the oxidation kinetics which obeys a logarithmic law at low temperatures or follows a parabolic law at high temperatures [21]. Starting from this last parabolic law, [2] derived an expression of oxygen flux:

$$(\Delta m)^2 = k_p t \quad \Rightarrow \quad \phi_{O_2} = \frac{d}{dt}(\Delta m)^2 = \frac{1}{2}\sqrt{\frac{k_p}{t}} \tag{1.68}$$

with Δm the mass gain and k_p the parabolic oxidation growth rate. Even if both types of boundary conditions have been implemented in the code (and verified in

[5]this definition assumes that the sample is prismatic

section 1.3), the second option will be preferred as they are more widely covered by scientific literature.

As explained previously, the experimental results are giving the time evolution of the oxide layer thickness. Our model currently predicts the distribution of the corrosion rate (i.e. the oxygen concentration) but not explicitly the oxide layer thickness. As the diffusion process is slow ($D \sim 10^{-6}$ µm s^{-2}), the oxidation reaction may be assumed to take place instantaneously [3], [22] and it exists henceforth a threshold beyond which the alloy matrix can be supposed completely oxidized. [3] suggested to take this upper bound limit as 65% of the boundary condition value (for example, if $c(x = 0, t) = c_0$ is imposed at the boundary, all points characterized by $c > 0.65c_0$ will be completely oxidized).

1.4.2 Results

The first set of data is taken from [1] who studied hot corrosion for hardened nickel based RR1000 alloy from 700°C to 800°C. For each temperature, the mass change was measured every 10h to reach the 200 hours of testing. The oxide layer thickness was then calculated assuming that 1 mg cm^{-2} represents[6] an increase by 6.08 µm. Thus, oxide layer depth was determined by calculations and not directly measured.

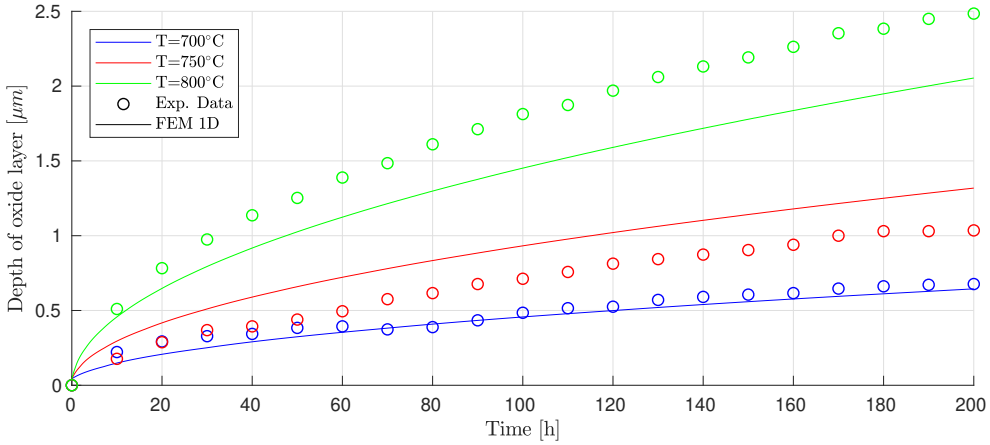


Figure 1.11: Comparison of numerical and experimental results taken from [1]. Parameters used for simulation in Table 1.2.

Figure 1.11 compares the numerical and experimental results for 700, 750 and 800°C. Parameters used in the simulation are displayed in Table 1.2. Unfortunately, the constants D have not been measured in [1], and consequently, chosen value of D have been taken from [2] who studied the same alloy at the same temperature. As it can be seen in Figure 1.11, the numerical curve fits the experimental data fairly well for the lowest temperature 700°C. However, the predictions are less accurate for higher temperatures, sometimes overestimating the actual layer depths, sometimes underestimating it. In all cases, the global square root time evolution is well captured, despite of the errors that reach more or less 30%. This indicates surely a bad parameters fitting (D in our case which

[6] based on atomic weight of oxygen and chromium and assuming that the oxidized matrix contains no pores (i.e. the calculated oxide layer thickness could be slightly underestimated)

Parameter	Value	Unit
$c(x,0)$	0	%
$c(0,t)$	100	%
c_{ox}	65	%
D $(T = 700°C)$	1.397×10^{-6}	$\mu m\, s^{-2}$
D $(T = 750°C)$	5.859×10^{-6}	$\mu m\, s^{-2}$
D $(T = 800°C)$	14.23×10^{-6}	$\mu m\, s^{-2}$
Δt	100	s
Δx	0.125	µm
`MaxTime`	200	h

Table 1.2: Parameters used for simulation Figure 1.11

regulates the propagation speed). We should not be surprised that the values of D picked from [2] do not match exactly those from [1] if they had been calculated. Indeed, as the experimental protocol is not exactly the same in both experiments, measurements of D are inevitably impacted.

Another way to convince the reader that the code predicts correctly the evolution of the oxide layer depth is to proceed to an *a posteriori* fitting with experimental data to determine D. Results shown in Figure 1.12 have been obtained with a such fitting for D: the diffusivities were determined to fit properly the first 5 experimental data ($0 < t < 40$h) for each temperature. As the behavior for $0 < t < 40$h is imposed by the fitting, observations that are made in this time region are skewed, and we must rather pay attention to experimental data with $t > 40$. Numerical predictions for $t > 40$h are rather accurate, indicating that, with a single suitable coefficient D, the corrosion propagation is well modeled. This also proves once more that the natural diffusion process is well adapted to the corrosion diffusion phenomenon.

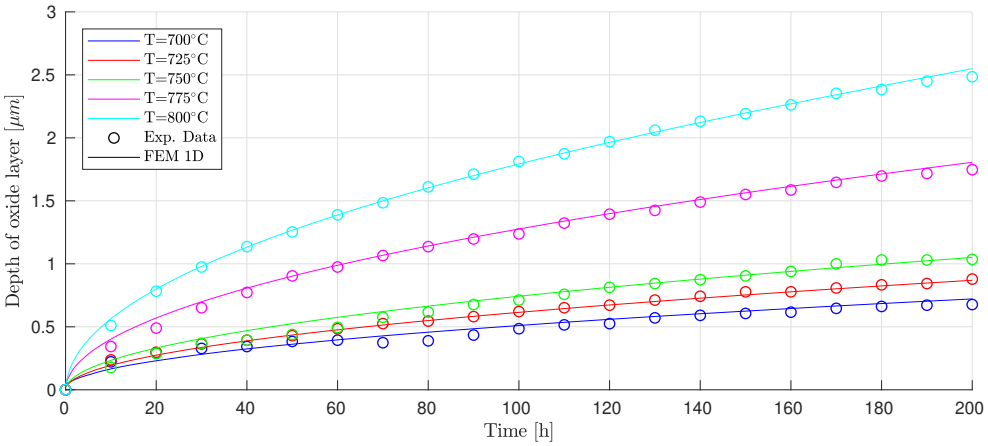


Figure 1.12: Comparison of numerical and experimental results taken from [1]. Parameters used for simulation in Table 1.2, and D for each temperature in Table 1.3 (a posteriori fitting for D)

T [°C]	D [µm s^{-2}]
700	1.75×10^{-6}
725	2.55×10^{-6}
750	3.73×10^{-6}
775	11.0×10^{-6}
800	21.7×10^{-6}

Table 1.3: Diffusion coefficient D determined by a posteriori fitting used in simulation Figure 1.12

1.4.3 About determination of diffusivity

The numerical curves are, as shown previously, relatively sensitive to the diffusion constant D whereas a certain uncertainty remains about its real value[7]. In order to exploit even more the available data, and discuss in detail the capabilities of the numerical model, several ways to determine the constant D have been investigated, based on the available data.

Several authors [21] proposed relationships linking the oxide layer thickness with time t and diffusivity D. In general, these relationships repose on the kinetics of the oxidation reaction. In the study case [1], oxidation kinetics is of second order and the relationship is given by the following equation

$$X^2 = Dt \tag{1.69}$$

with X the oxide layer thickness (including both external and internal damage) that can be measured from F.I.B. microscopy. Since F.I.B. analysis provides local rather than global value for X, it is necessary to perform several measurements of X all along the sample surface, to approach an average value for X (and D consequently). Instead of an averaged value of X, it may be preferred to seek for the maximal X, as it constitutes the most conservative value. Study [1] has determined from F.I.B. microscopy maximal values of X after 200 hours at 700, 750 and 800°C. Starting from these values, the corresponding diffusion coefficients D have been calculated with (1.69) (values are exposed in Table 1.4).

T [°C]	X [µm]	D [µm s^{-2}]
700	1.45	2.901×10^{-6}
750	2.1	6.125×10^{-6}
800	3.9	21.13×10^{-6}

Table 1.4: Upper values of internal oxidation depth from FIB examination (taken from [1]) and corresponding diffusion coefficients D, for different temperatures

The numerical results superimposed with experimental data are shown in Figure 1.13. The 3 numerical curves experience the same global shape but are not completely in accordance with experimental data, by overestimating the real oxide layer thickness. The overestimation arises especially for $T = 700$°C and 750°C.

[7] D being function of the surface state [23], the alloy composition and thus the oxide content [3] but is also strongly influenced by temperature, etc.

For 700°C for example, the numerical solution predicts an oxide layer 30% thicker as it is, after 200 hours. Compared with the previous simulation (Figure 1.11), diffusion coefficient D is more than twice the value referenced by [2]. Consequently, as the numerical model strongly depends on the diffusivity, a certain non negligible error can be induced by an inadequately chosen constant D. If a maximal value for X is considered, it must be kept in mind that this will result in a overestimation of the oxide layer thickness.

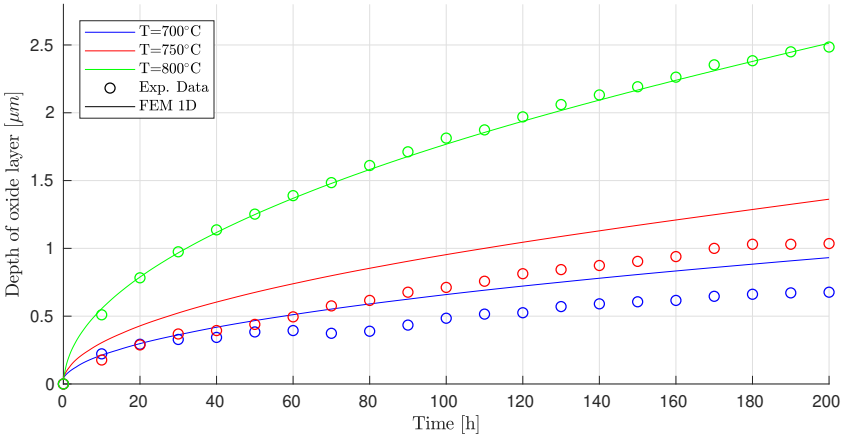


Figure 1.13: Comparison of numerical and experimental results taken from [1]. Parameters used for simulation in Table 1.2, and D for each temperature in Table 1.4.

It has been said before that D is strongly dependent of the working temperature. Authors [24] and [21] suggested that a standard relationship between D and temperature T was given by Arrhenius equation:

$$D(T) = D_0 \cdot \exp\left(\frac{-Q}{RT}\right) \tag{1.70}$$

where Q is the activation energy, R the universal gas constant and D_0 a constant. [24] warned however that this equation should be considered as a basic standard rather than a usual generality. This last equation provides indeed a convenient way to extrapolate value of D for different temperatures, as long as D_0 and Q are known. For example, experimental data for $T = 725$°C are available in [1] but D is not known at this temperature. Using D at 700°C as it was done in Figure 1.11 (which has led to a good fitting), we can calculate D_0 and subsequently extrapolate D from (1.70) at chosen temperature. Extrapolated values of D are given in Table 1.5 and corresponding numerical results are shown in Figure 1.14. Results for 700°C and 750°C are overevaluated by about 30% and are more accurate for 800°C. As stated by [24], diffusivity extrapolation from (1.70) gives indicative yet not perfectly reliable outputs as it may result in an overestimation.

Finally, it must be held in mind that the numerical model does not account for material variability in its current development: D is kept constant everywhere and at all times. This is of course never observed in reality as the oxygen diffusivity is larger in oxides products than in alloy matrix. Keeping D constant over both oxidized and sane matrix will hence lead to an underestimation of oxide layer thickness and therefore to unconservative results. No paper about study

T [°C]	D [µm s^{-2}]
725	3.2238×10^{-6}
750	7.1415×10^{-6}
775	15.231×10^{-6}
800	31.358×10^{-6}

Table 1.5: Extrapolated values of D from (1.70) for different temperatures

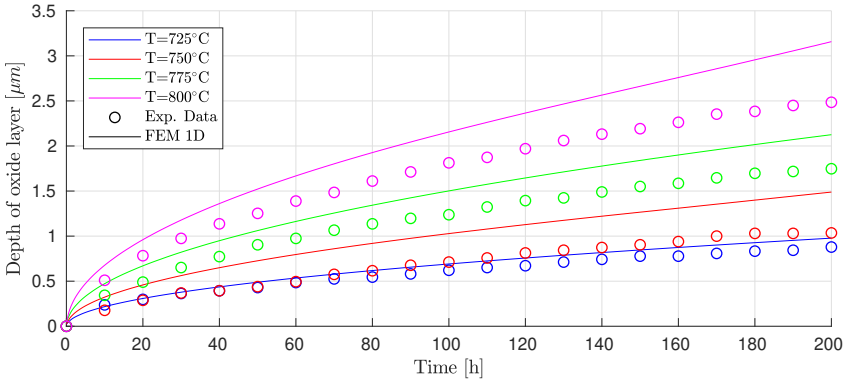


Figure 1.14: Comparison of numerical and experimental results taken from [1]. Parameters used for simulation in Table 1.2, and D for each temperature in Table 1.5.

of diffusivity in oxidized matrix have been found in the literature. [3] considered all the same two distinct diffusivities D_1 and D_2 in his model for oxidized (or non-oxidized) domains. Even if the lack of reference for diffusivity in oxidized matrix prevents to perform a relevant comparison between experimental data and numerical results (generated by the code involving 2 distinct constants D_1 and D_2), this functionality has nevertheless been developed in the code.

1.5 Conclusion

As a conclusion of this chapter dedicated to the first one-dimensional model, some conclusion must be retained.

The code developed numerically assumed that corrosion diffusion was a **purely diffusive phenomenon**. The diffusivity D was kept constant in the equations development, even if the code can accommodate a non-uniform distribution of D (for example for oxidized domain and alloy matrix).

Regarding the verification, it has been shown that

- Refining the mesh lead to a smaller error and then a more accurate solution.
- Even for coarse meshes, the numerical solution provides acceptable results. Results become truly accurate from 10 elements, even if the situation involves brutal changes in c.

- The largest errors are experienced at the instant following immediately the initial time, and more generally anywhere where the gradient of corrosion is high. This error can be decreased by opting for a finer mesh rather than decreasing the time step. Moreover, refining the time step has not led to a smaller error, indicating that the time integrating scheme is not questioned. The problem was rather due to finite element method who enforce the weak form to be satisfied on average in an element, and if a coarse mesh was chosen, the numerical model was not sufficiently reach to accommodate high gradients of c (and even less so for brutal discontinuities such as imposed at the beginning of the first test case). Finally these errors **do not affect the quality** of the approximation solution for larger values of time (i.e. these errors are **relative** and not **absolute** and are consequently not accumulated).

- The code can model essential as well as natural boundary conditions.

Based on of these considerations, the code is considered as verified.

Regarding the validation step, it was firstly investigated whether essential or natural boundary conditions should be considered to model the inflow of oxygen into the solid. Only essential boundary conditions have been considered in the validation step.

In order to compare our numerical predictions with experimental data, it has been assumed that the alloy matrix was fully oxidized as soon as the corrosion rate c reached 65% of the value c_0 at the boundary. According to the transient diffusion pattern (the closer to the boundaries, the higher the corrosion rate), this assumptions induces a systematic splitting of the solid into two domains: one fully oxidized and one alloy matrix. The diffusivity is not the constant across both domains but was assumed to be in the validation process (the considered experiment was not providing value of D in oxidized matrix). However, this feature has been implemented in the code (it can thus accommodate non uniform distribution of D).

The present analysis finally proved that the **natural diffusion equation** developed in this model **was appropriately chosen** to predict the **oxide layer depth time evolution** provided that the constant D was known. Several difficulties were experienced to match completely the experimental data as no perfectly consistent values for D were provided in the considered experiment. Hence, the question of the determination of the diffusion coefficient D was investigated.

Several methods for determination of D have been approached in the previous section. One of them resulted in a conservative overestimation of D and subsequently an overestimation of the oxide layer thickness. This was likely caused by FIB microscopy analysis protocol that was only retaining the **maximal values of depths of internal oxidation** ($= X$). More realistic (but less conservative) values of D could be approached by considering **averaged values of depths of internal oxidation** (not communicated in the reviewed experiments). Arrhenius equation was used to extrapolated D at different temperatures and has led to rather uncertain results providing sometimes fairly accurate results, sometimes overestimates by about 30%.

Chapter 2

Two-Dimensional and transient corrosion diffusion model

This chapter is dedicated to the extension of the first model to a two dimension space.

2.1 Derivation of the differential equation

As it has been done previously, it will be assumed, in accordance with [2], that the corrosion diffusion follows a purely diffusive law.

$$\frac{\partial c}{\partial t} = \nabla \cdot (D\nabla c - MDc\nabla p) \tag{2.1}$$

If the diffusivity D is constant and the pressure factor is neglected, one have

$$\frac{\partial c}{\partial t} = D\nabla^2 c = D\Delta c \tag{2.2}$$

Or alternatively,

$$\frac{\partial c}{\partial t} = D\left(\frac{\partial^2 c}{\partial x^2} + \frac{\partial^2 c}{\partial y^2}\right) \tag{2.3}$$

Once more, this last equation is a purely natural diffusive partial differential equation and represents the propagation of the corrosion inside the material.

2.2 Solving of the partial differential equation by finite elements method

2.2.1 Derivation of the weak form of the problem

The weak form of the problem can be derived using the weighted residual method. An approximation function $\tilde{c}(x, y) \approx c(x, y)$ will be looked for. The weighted

residual is formed by considering this approximation function and introducing one weighting function $\delta\tilde{c}(x,y)$, and integrating over a studied volume

$$\texttt{WR} = \int_V D \cdot \left(\frac{\partial^2 \tilde{c}}{\partial x^2} + \frac{\partial^2 \tilde{c}}{\partial y^2} \right) \cdot \delta\tilde{c} \; dV - \int_V \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c} \; dV \tag{2.4}$$

The thickness e of the considered plate is supposed to be constant, the integration over the volume gives the following expression

$$\Leftrightarrow \texttt{WR} = eD \cdot \int_\Omega \left(\frac{\partial^2 \tilde{c}}{\partial x^2} + \frac{\partial^2 \tilde{c}}{\partial y^2} \right) \cdot \delta\tilde{c} \; d\Omega - e \cdot \int_\Omega \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c} \; d\Omega \tag{2.5}$$

The Green-Ostrogradski's theorem enables to transform the first integral over a surface in (2.5) into a line integral performed over the surface boundaries plus an integral over the same surface:

$$\int_\Omega \phi \frac{\partial \psi}{\partial x} \; d\Omega = \oint_\Gamma (\phi\psi) \; d\Gamma - \int_\Omega \psi \frac{\partial \phi}{\partial x} \; d\Omega$$

Applied to (2.5), with $\psi = \frac{\partial \tilde{c}}{\partial x}$ and $\phi = \delta\tilde{c}$, this gives

$$\texttt{WR} = eD \cdot \oint_\Gamma \left(\frac{\partial \tilde{c}}{\partial x} + \frac{\partial \tilde{c}}{\partial y} \right) \cdot \delta\tilde{c} \; d\Gamma - eD \cdot \int_\Omega \left(\frac{\partial \tilde{c}}{\partial x} \cdot \frac{\partial \delta\tilde{c}}{\partial x} + \frac{\partial \tilde{c}}{\partial y} \cdot \frac{\partial \delta\tilde{c}}{\partial y} \right) \; d\Omega - e \int_\Omega \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c} \; d\Omega \tag{2.6}$$

Finally, the weighted residual WR is equaled to zero.

$$D \cdot \oint_\Gamma \left(\frac{\partial \tilde{c}}{\partial x} + \frac{\partial \tilde{c}}{\partial y} \right) \cdot \delta\tilde{c} \; d\Gamma - D \cdot \int_\Omega \left(\frac{\partial \tilde{c}}{\partial x} \cdot \frac{\partial \delta\tilde{c}}{\partial x} + \frac{\partial \tilde{c}}{\partial y} \cdot \frac{\partial \delta\tilde{c}}{\partial y} \right) \; d\Omega - \int_\Omega \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c} \; d\Omega = 0 \tag{2.7}$$

Equation (2.8) defines the weak form of the problem. The first term represents the boundary conditions while the second and third terms describe the diffusion phenomenon respectively in terms of space (inside the studied space) and in terms of time.

For convenience, the boundary conditions term of (2.7) will be treated apart in section 2.2.6, assuming here that it has no influence on the weighted residual. In other words, the weak form will be considered as:

$$- D \cdot \int_\Omega \left(\frac{\partial \tilde{c}}{\partial x} \cdot \frac{\partial \delta\tilde{c}}{\partial x} + \frac{\partial \tilde{c}}{\partial y} \cdot \frac{\partial \delta\tilde{c}}{\partial y} \right) \; d\Omega - \int_\Omega \frac{\partial \tilde{c}}{\partial t} \cdot \delta\tilde{c} \; d\Omega = 0 \tag{2.8}$$

2.2.2 Derivation of the expressions of local stiffness matrix and local nodal load vector

Everywhere inside a finite element, the continuous scalar field $c(x,y)$ is approximated by $\tilde{c}(x,y)$, which can be expressed as a linear combination of a chosen set of shape functions:

$$c(x,y) \approx \tilde{c}(x,y) = \sum_{i=1}^{n_{dof}} N_i(x,y) \cdot q_i \tag{2.9}$$

Where $N_i(x, y)$ represents the shape function associated to degree of freedom i, while q_i is the nodal unknown associated to the same degree of freedom. Note that n_{dof} refers to the number of degree of freedom of the considered element. This equation may be rewritten on a vectorial form

$$\tilde{c}(x, y) = \mathbf{N}\mathbf{q} \tag{2.10}$$

The shape function vector $\mathbf{N}$ is a 1-by-n_{dof} vector while $\mathbf{q}$ is a column vector with as many components. Subsequently:

$$\mathbf{N} = \begin{bmatrix} N_1(x) & N_2(x) & \cdots & N_{n_{dof}}(x) \end{bmatrix}, \qquad \mathbf{q} = \begin{bmatrix} q_1 \\ q_2 \\ \cdots \\ q_{n_{dof}} \end{bmatrix}$$

Before substituting the approximation function $\tilde{c}(x, y)$ in the weak form, several additional quantities need to be expressed as a function on these $N_i(x, y)$. The vector weighting function $\delta\tilde{c}(x, y)$ can be expressed as

$$\delta\tilde{c}(x, y) = \sum_{i=1}^{n_{dof}} N_i(x, y) \cdot \delta q_i = \mathbf{N} \cdot \delta\mathbf{q} \tag{2.11}$$

From (2.9), it is easy to prove that

$$\frac{\partial\tilde{c}}{\partial x} = \sum_{i=1}^{n_{dof}} \frac{\partial\tilde{N}_i}{\partial x} \cdot q_i \qquad\qquad \frac{\partial\tilde{c}}{\partial y} = \sum_{i=1}^{n_{dof}} \frac{\partial\tilde{N}_i}{\partial y} \cdot q_i \tag{2.12}$$

Once more, the vectorial formalism will be preferred,

$$\frac{\partial\tilde{c}}{\partial x} = \mathbf{B}_x\mathbf{q} \qquad\qquad \frac{\partial\tilde{c}}{\partial y} = \mathbf{B}_y\mathbf{q} \tag{2.13}$$

where

$$B_{x,i} = \frac{\partial N_i}{\partial x} \qquad\qquad B_{y,i} = \frac{\partial N_i}{\partial y} \tag{2.14}$$

It can be proven in the same way that

$$\frac{\partial\delta\tilde{c}}{\partial x} = \mathbf{B}_x\delta\mathbf{q} \qquad\qquad \frac{\partial\delta\tilde{c}}{\partial y} = \mathbf{B}_y\delta\mathbf{q} \tag{2.15}$$

And because $\tilde{\delta}c$ is a scalar quantity: $\delta\tilde{c}^T = \delta\tilde{c}$ and $\tilde{c}^T = \tilde{c}$ and subsequently : $\delta\mathbf{q}^T\mathbf{N}^T = \mathbf{N}\delta\mathbf{q}$ and $\mathbf{q}^T\mathbf{N}^T = \mathbf{N}\mathbf{q}$. Substituting all these previous quantities into the weak form (2.8) gives the following:

$$-D\int_\Omega \mathbf{B}_x\mathbf{q}\mathbf{B}_x\delta\mathbf{q} + \mathbf{B}_y\mathbf{q}\mathbf{B}_y\delta\mathbf{q}\, d\Omega - \frac{\partial}{\partial t}\int_\Omega \mathbf{N}\mathbf{q}\mathbf{N}\delta\mathbf{q}\, d\Omega = 0 \tag{2.16}$$

$$\Leftrightarrow -\delta\mathbf{q}^T\left(D\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y\, d\Omega\right)\mathbf{q} - \delta\mathbf{q}^T\frac{\partial}{\partial t}\int_\Omega \mathbf{N}^T\mathbf{N}\mathbf{q}\, d\Omega = 0 \tag{2.17}$$

The time derivative can be derived using first order backward finite difference:

$$\frac{\partial f}{\partial t} = \frac{f^t - f^{t-\Delta t}}{\Delta t} \tag{2.18}$$

Because $\mathbf{N}$ are only functions of x, the single variable affected by the time derivative are the nodal unknowns $\mathbf{q}$.

$$\Leftrightarrow -D\delta\mathbf{q}^T\left(\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y\, d\Omega\right)\mathbf{q}^t - \delta\mathbf{q}^T\int_\Omega \mathbf{N}^T\mathbf{N}\left(\frac{\mathbf{q}^t-\mathbf{q}^{t-1}}{\Delta t}\right)\, d\Omega = 0 \tag{2.19}$$

With several operations,

$$\delta\mathbf{q}^T\left(-\int_\Omega D\Delta t\mathbf{B}_x^T\mathbf{B}_x + D\Delta t\mathbf{B}_y^T\mathbf{B}_y + \mathbf{N}^T\mathbf{N}\, d\Omega\right)\mathbf{q}^t + \delta\mathbf{q}^T\int_\Omega \mathbf{N}^T\mathbf{N}\, d\Omega\, \mathbf{q}^{t-1} = 0 \tag{2.20}$$

Noting that $\delta\mathbf{q}$ contains only arbitrary components, the equation may be simplified as follows

$$\left(-\int_\Omega D\Delta t\mathbf{B}_x^T\mathbf{B}_x + D\Delta t\mathbf{B}_y^T\mathbf{B}_y + \mathbf{N}^T\mathbf{N}\, d\Omega\right)\mathbf{q}^t + \int_\Omega \mathbf{N}^T\mathbf{N}\, d\Omega\, \mathbf{q}^{t-1} = 0 \tag{2.21}$$

This last equation highlights the expressions of stiffness matrix and nodal loads vector. This expression may then be rewritten as

$$\mathbf{K}_e\, \mathbf{q}_e^t = \mathbf{g}_e^t \tag{2.22}$$

With $\mathbf{K}_e$ the elementary stiffness matrix and $\mathbf{g}$ the elementary consistent nodal loads vector whose expressions are given by

$$\mathbf{K}_e = \int_\Omega D\Delta t\mathbf{B}_x^T\mathbf{B}_x + D\Delta t\mathbf{B}_y^T\mathbf{B}_y + \mathbf{N}^T\mathbf{N}\, d\Omega \tag{2.23}$$

$$\mathbf{g}_e^t = \int_\Omega \mathbf{N}^T\mathbf{N}\, d\Omega\, \mathbf{q}^{t-1} \tag{2.24}$$

Equation (2.22) shows that an approximating solution of (2.3) can be found by solving a linear system where the independent term changes at each iteration (is dependent of the solution $\mathbf{q}$ at previous step).

2.2.3 Derivation of the expressions of global stiffness matrix and global load vector

Now that the expressions of local stiffness matrix and local nodal consistent load vector have been derived, the studied domain can be extended to a mesh of finite elements. The studied domain can be discretized in finite elements whose stiffness matrices and nodal loads vectors are known. To build the global stiffness matrix $\mathbf{K}$ knowing each local stiffness matrix $\mathbf{K}_e$, it is necessary to perform an assembling operation over each $\mathbf{K}_e$. The same operation is executed to get $\mathbf{g}^t$

$$\mathbf{K} = \mathop{\mathrm{A}}_{el=1}^{N_{el}} \mathbf{K}_e = \mathop{\mathrm{A}}_{el=1}^{N_{el}} D\int_\Omega \Delta t\mathbf{B}_x^T\mathbf{B}_x + \Delta t\mathbf{B}_y^T\mathbf{B}_y + \mathbf{N}^T\mathbf{N}\, d\Omega \tag{2.25}$$

$$\mathbf{g}^t = \mathop{\mathrm{A}}_{el=1}^{N_{el}} \mathbf{g}_e = \mathop{\mathrm{A}}_{el=1}^{N_{el}} \left(\int_\Omega \mathbf{N}^T\mathbf{N}\, d\Omega\, \mathbf{q}^{t-1}\right) \tag{2.26}$$

By solving the following linear system for $\mathbf{q}^t$

$$\mathbf{K}\mathbf{q}^t = \mathbf{g}^t$$

the approximating solution $\tilde{c}(x,y)$ of equation (2.3) can be found all across the domain considering

$$\tilde{c}(x,y) = \mathbf{N}\mathbf{q}^t$$

2.2.4 Shape functions

As the weak form of the problem (2.8) involves first order spatial derivative, the shape functions used to build the approximation must be at least linear. Therefore, two dimensional finite elements with 4 nodes[1] (= quad4 element, see Figure 2.1-(a)) are used.

To describe the shape function, only the reduced dimensionless coordinates system will be used in order to facilitate the integration of the local stiffness matrices. The associated elementary shape functions expressed in the dimensionless coordinates system are given by

$$N_1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta)$$
$$N_2(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta)$$
$$N_3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta)$$
$$N_4(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta)$$

With $\xi \in [-1,\ 1]$ and $\eta \in [-1,\ 1]$. These shape functions are represented in Figure 2.1.

Still in the same dimensionless coordinate system, the first derivative of the shape functions matrix $\mathbf{N}$ with respect to the global variables x and y can be derived using chain rule derivative,

$$B_{x,i}(\xi,\eta) = \frac{\partial N_i}{\partial x} = \frac{\partial N_i}{\partial \xi}\cdot\frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta}\cdot\frac{\partial \eta}{\partial x} \tag{2.27}$$

$$B_{y,i}(\xi,\eta) = \frac{\partial N_i}{\partial y} = \frac{\partial N_i}{\partial \xi}\cdot\frac{\partial \xi}{\partial y} + \frac{\partial N_i}{\partial \eta}\cdot\frac{\partial \eta}{\partial y} \tag{2.28}$$

In these last two equations, the first derivatives of N_i can be directly calculated from the definition of N_i:

$$\frac{\partial N_1}{\partial \xi} = -\frac{1}{4}(1-\eta) \qquad \frac{\partial N_1}{\partial \eta} = -\frac{1}{4}(1-\xi) \tag{2.29}$$

$$\frac{\partial N_2}{\partial \xi} = -\frac{1}{4}(1+\eta) \qquad \frac{\partial N_2}{\partial \eta} = \frac{1}{4}(1-\xi) \tag{2.30}$$

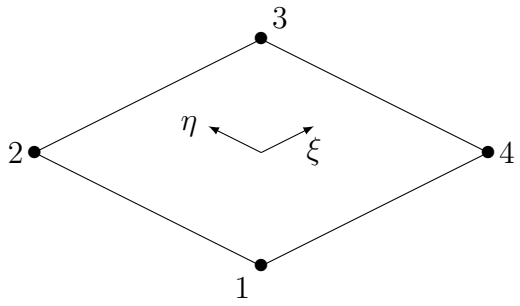
$$\frac{\partial N_3}{\partial \xi} = \frac{1}{4}(1+\eta) \qquad \frac{\partial N_3}{\partial \eta} = \frac{1}{4}(1+\xi) \tag{2.31}$$

$$\frac{\partial N_4}{\partial \xi} = \frac{1}{4}(1-\eta) \qquad \frac{\partial N_4}{\partial \eta} = -\frac{1}{4}(1+\xi) \tag{2.32}$$

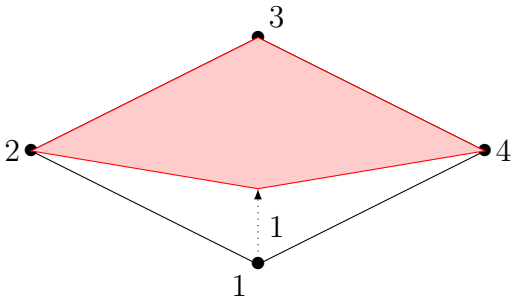
In order to ease the readability, the elementary matrices $\mathbf{B}_\xi$ and $\mathbf{B}_\eta$ are also introduced

$$\mathbf{B}_\xi = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \end{bmatrix} \qquad \mathbf{B}_\eta = \begin{bmatrix} \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix}$$

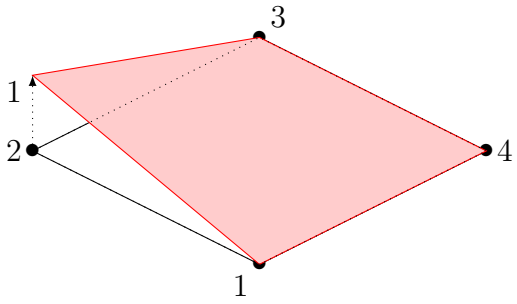
[1] It means then $n_{dof} = 4$ in (2.9)



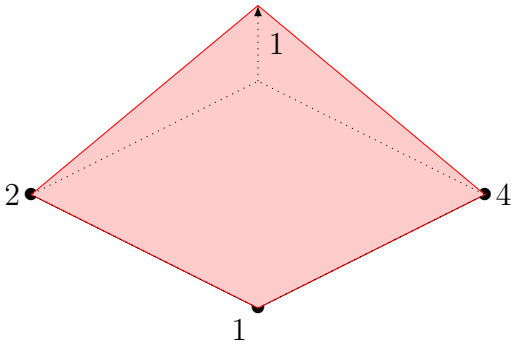
(a) Quad4 isoparametric element and local numbering



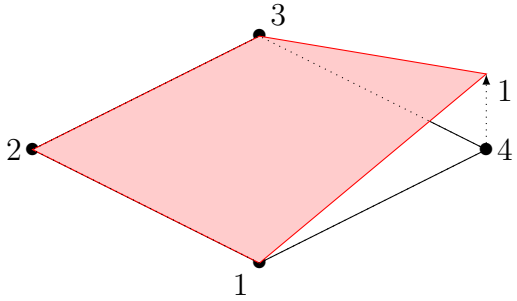
(b) Shape function 1 : $N_1(\xi, \eta)$



(c) Shape function 2 : $N_2(\xi, \eta)$



(d) Shape function 3 : $N_3(\xi, \eta)$



(e) Shape function 4 : $N_4(\xi, \eta)$

Figure 2.1: Shape functions associated to an element of the mesh

The second factor of each term of (2.27) and (2.28)($\frac{\partial \eta}{\partial(.)}$ and $\frac{\partial \xi}{\partial(.)}$) are not directly known, but can be evaluated using the jacobian matrix of the regular change of variable. Indeed, starting from the basic principle of mapped finite element, x and y inside a considered element are expressed as a linear combination of shape functions and adjacent nodal coordinates x_i and y_i ($i = 1, \ldots, 4$).

$$\begin{cases} x = \sum_{i=1}^{4} N_i(\xi, \eta) \cdot x_i = f(\xi, \eta) \\ y = \sum_{i=1}^{4} N_i(\xi, \eta) \cdot y_i = g(\xi, \eta) \end{cases} \tag{2.33}$$

By definition of a total derivative:

$$\begin{cases} dx = \dfrac{\partial x}{\partial \xi} d\xi + \dfrac{\partial x}{\partial \eta} d\eta \\ dy = \dfrac{\partial y}{\partial \xi} d\xi + \dfrac{\partial y}{\partial \eta} d\eta \end{cases} \tag{2.34}$$

Or alternatively,

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \underbrace{\begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix}}_{=\mathbf{J}} \begin{bmatrix} d\xi \\ d\eta \end{bmatrix} \tag{2.35}$$

The matrix on the right hand side defines the jacobian matrix $\mathbf{J}$. The 4 components of the matrix are given by

$$\frac{\partial x}{\partial \xi} = \frac{\partial \mathbf{N}}{\partial \xi} \cdot \mathbf{x} = \mathbf{B}_\xi \cdot \mathbf{x} \qquad \frac{\partial x}{\partial \eta} = \frac{\partial \mathbf{N}}{\partial \eta} \cdot \mathbf{x} = \mathbf{B}_\eta \cdot \mathbf{x}$$

$$\frac{\partial y}{\partial \xi} = \frac{\partial \mathbf{N}}{\partial \xi} \cdot \mathbf{y} = \mathbf{B}_\xi \cdot \mathbf{y} \qquad \frac{\partial y}{\partial \eta} = \frac{\partial \mathbf{N}}{\partial \eta} \cdot \mathbf{y} = \mathbf{B}_\eta \cdot \mathbf{y}$$

The inverse $\mathbf{J}^{-1}$ of the jacobian matrix contents the 4 searched partial derivatives:

$$\mathbf{J}\mathbf{J}^{-1} = \mathbb{1} \quad \Rightarrow \mathbf{J}^{-1} = \begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \xi}{\partial y} \\ \dfrac{\partial \eta}{\partial x} & \dfrac{\partial \eta}{\partial y} \end{bmatrix} \tag{2.36}$$

Combining (2.27), (2.28) and (2.36), the first derivatives $\mathbf{B}_x$ and $\mathbf{B}_y$ of the shape function matrix $\mathbf{N}$ can now be determined.

Using dimensionless coordinate system with variables ξ and η to evaluate the integral, the local expression of the elementary stiffness matrix (2.23) can be expressed as follow

$$\mathbf{K}_e = D\Delta t \int_{-1}^{1} \int_{-1}^{1} \left(\mathbf{B}_x^T \mathbf{B}_x + \mathbf{B}_y^T \mathbf{B}_y + \mathbf{N}^T \mathbf{N} \right) \cdot J \, d\xi \, d\eta \tag{2.37}$$

Where J is the jacobian of the regular change of variable. These operations are repeated to get $\mathbf{g}_e^t$:

$$\mathbf{g}_e^t = \int_{-1}^{1} \int_{-1}^{1} \mathbf{N}^T \mathbf{N} \cdot J \, d\xi \, d\eta \, \mathbf{q}_e^{t-1} \tag{2.38}$$

2.2.5 Numerical Integration

The evaluation of local stiffness matrices and nodal load vectors through equations (2.37) and (2.38) require both the calculation of a surface integral. As the expressions of the integrant can be slightly complicated, it is that time highly relevant to relinquish the analytical development and perform a numerical integration. As it has been done previously in section 1, the Gaussian quadrature

will be chosen. Extended to 2 dimensions, this method states that a function of 2 independent variables $f(\xi, \eta)$ can be integrated as follow

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\, d\xi\, d\eta \approx \sum_{i=1}^{n}\sum_{j=1}^{n} H_i^* H_j^* \cdot f\left(\xi_j^*, \xi_i^*\right) \tag{2.39}$$

This equation is easily demonstrated recalling from 1D that:

$$I_1(\eta) = \int_{-1}^{1} f(\xi,\eta)\, d\xi \approx \sum_{i=1}^{n} H_i^* \cdot f(\xi_i^*, \eta) \tag{2.40}$$

During the calculation of this integral, the variable η is treated as a constant. I_1 becomes then a single variable function and I is finally given by the integral

$$I = \int_{-1}^{1} I_1(\eta) d\eta \approx \sum_{j=1}^{n} H_j^* \cdot I_1(\eta_j^*) \tag{2.41}$$

Substituting the expression of I_1 into 2.41, the equation 2.39 is well recovered.

To keep the vision of a surface integral rather than two successive integrals, equation (2.39) can be rewritten as

$$I = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta)\, d\xi\, d\eta = \sum_{i=1}^{n} H_i \cdot f\left(\xi_i^*, \eta_i^*\right) \tag{2.42}$$

Where ξ_i^* and η_i^* (with $i = 1, \ldots, n$) are n couples of coordinates defining the location of the n Gaussian in the two dimensional space. This formulation is exactly the same as the one used in section 1.2.6.

Regarding the precision of the integration, it is known by [25] that for linear shape function, a minimum of 2 Gauss points ($n = 2$) are required to perform an exact calculation of the integral following equation (2.39).

If the second formulation (2.42) is preferred, $2^2 = 4$ Gauss points are required the perform an exact calculation of the surface integral stated by (2.37) and (2.38). These 4 Gauss points are shown in Figure 2.2 and their coordinates and associated weights are detailed in Table 2.1.

Gauss Point	Abscissa: ξ_i^* [-]	Ordinate: η_i^* [-]	Weight [-]
G_1	$-\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	1
G_2	$-\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	1
G_3	$\frac{1}{\sqrt{3}}$	$\frac{1}{\sqrt{3}}$	1
G_4	$\frac{1}{\sqrt{3}}$	$-\frac{1}{\sqrt{3}}$	1

Table 2.1: Coordinates of Gauss points and associated weights

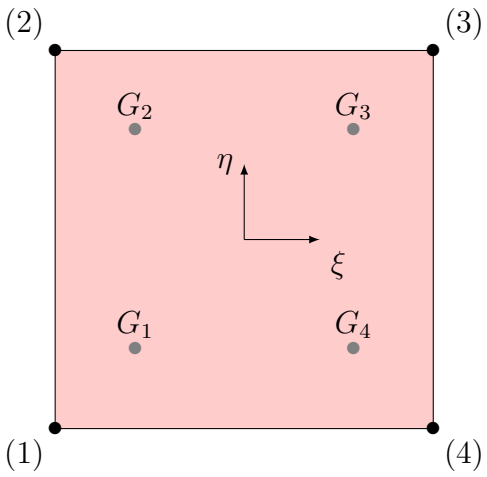


Figure 2.2: Gauss points representation inside a mapped element in reduced coordinates system

2.2.6 Boundary Conditions

As this section extends the remarks previously made in 1.2.7 for the 2D case, only the mains ideas will be summarized and the reader is invited to refer to 1.2.7 for additional justifications.

The main novelty in 2D is that boundary conditions, no matter their nature, may be applied on both nodes and lines: both features are handled by the code.

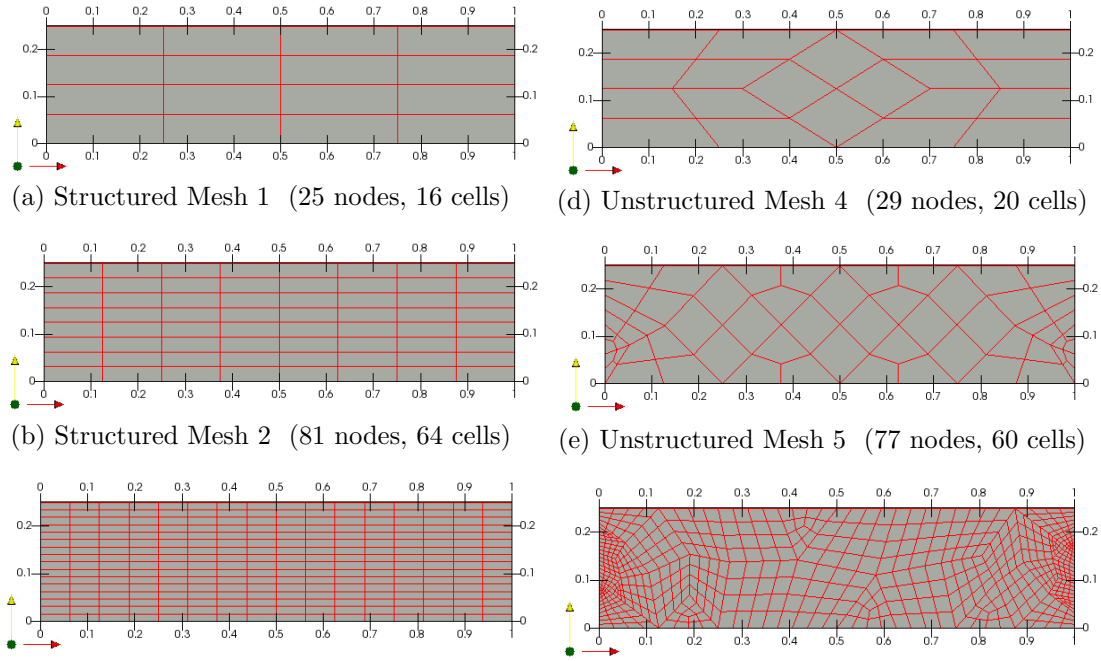
Boundary can be either applied on c or on its derivative, defining respectively essential and natural boundary conditions. Essential boundary conditions are directly applied to the generalized displacements vector $\mathbf{q}^t$ (influencing the nodal load vector at time $t+1$). For natural boundary conditions, fluxes are directly added to the nodal load vector after being multiplied by Δt. For fluxes applied on line elements, a consistent nodal load vector is first determined and then treated similarly as a set of nodal loads:

$$\mathbf{K}_e \mathbf{q}_e = \mathbf{g}_e^t + D\Delta t \boldsymbol{\phi}_c \tag{2.43}$$

2.3 Model verification

Since the diffusion takes now place in 2D, it becomes unreasonable to seek for an analytical solution as reference solution. Alternatively, it is proposed to analyze the results along a line parallel or perpendicular to x or y axis, and see if the results from 1D model are recovered. The verification step is thus conducted on the same basis than for 1D model, and for the same test cases, adapted to 2D space. Direct comparison between 1D and 2D are only possible if corrosion flux or corrosion rate are uniformly distributed along one direction y (or x): in that case, there is no diffusion in y direction, while diffusion in x takes place uniformly over the height of the plate (diffusion in x and y are thus decoupled).

All the cases presented below have been tested on structured and unstructured meshes. Figure 2.3 presents the different meshes used in this section.



(a) Structured Mesh 1 (25 nodes, 16 cells)

(b) Structured Mesh 2 (81 nodes, 64 cells)

(c) Structured Mesh 3 (289 nodes, 256 cells)

(d) Unstructured Mesh 4 (29 nodes, 20 cells)

(e) Unstructured Mesh 5 (77 nodes, 60 cells)

(f) Unstructured Mesh 6 (665 nodes, 600 cells)

Figure 2.3: Used meshes

2.3.1 Test Case 1

The situation is summarized in the Figure 2.4: corrosion rate of 100% is applied at each lateral edges of the plate, while an initial rate of 0% is initially assigned to the entire body.

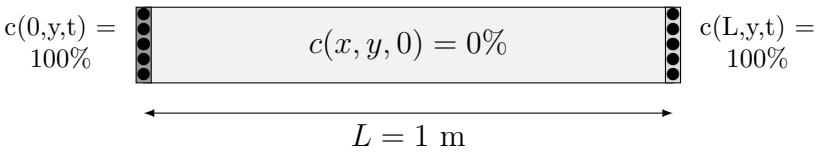
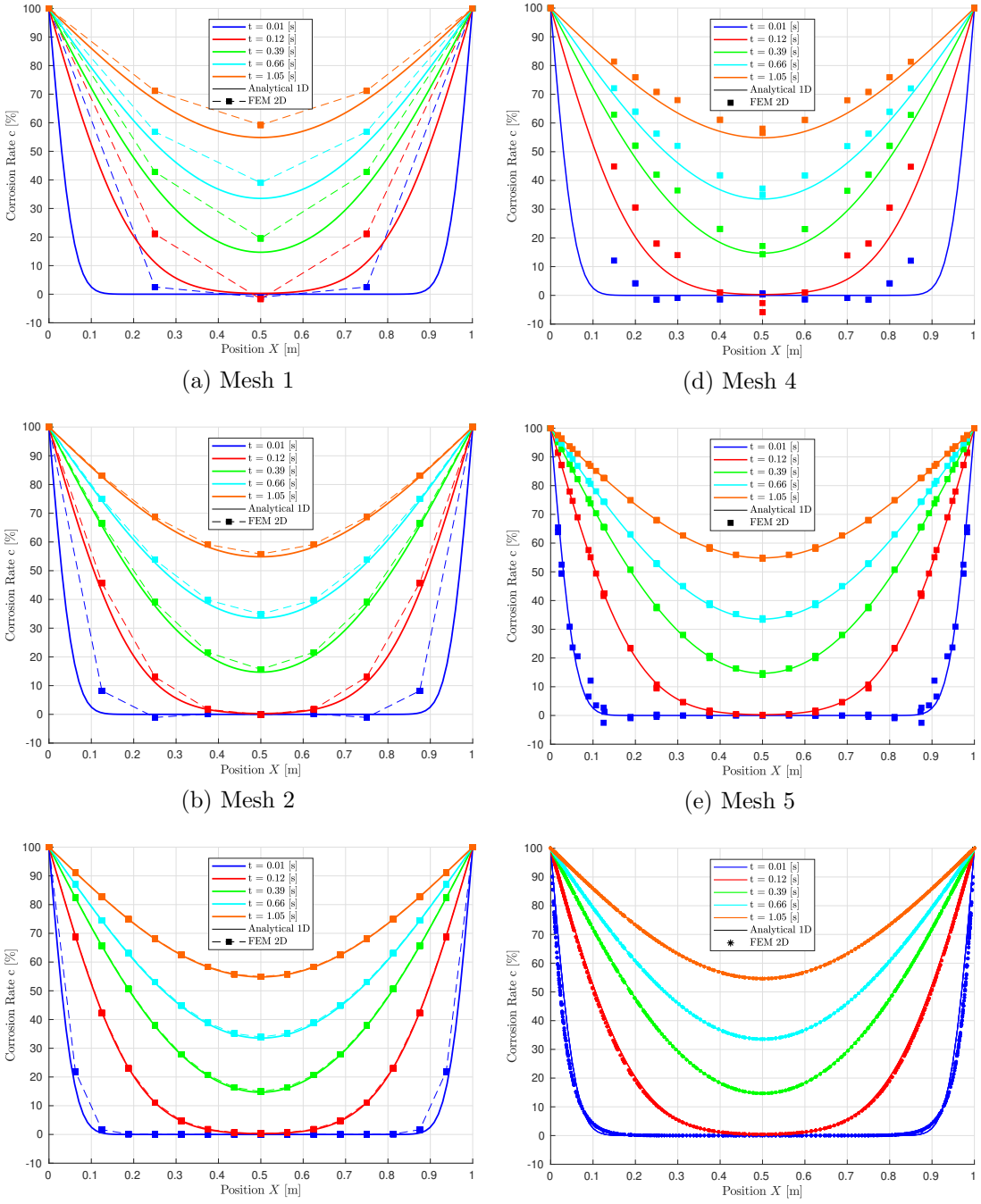


Figure 2.4: Initial and boundary conditions for study case 1

Numerical results obtained from the code are shown in Figure 2.5 for the 6 considered meshes. All nodal values for c are plotted at the right x position (corresponding the node coordinates), but **indifferently from their y position**. Figure 2.5 (a),(b) and (c) describes the behavior for structured meshes and shows that all nodes located at the same x coordinates have identical nodal values (square markers are superimposed). This indicates that the results are uniform over the height of the plate (y direction), which was expected since the boundary conditions are also uniformly imposed on y. Thus, the 2D code does not generates parasitic diffusion into y direction. However, Figure 2.5 (d)-(e)-(f) does not show superimposed nodes anymore: this is justifiable noting that unstructured meshes



(a) Mesh 1

(d) Mesh 4

(b) Mesh 2

(e) Mesh 5

(c) Mesh 3

(f) Mesh 6

Figure 2.5: Results obtained from 2D-diffusion code, superimposed indifferently of their y position for each 6 meshes considered in test case 1 ((parameters: $\Delta t = 0.01$ s, $D = 0.5$ m^2 s^{-1}, $c(L, y, t) = 100\%$, $c(x, y, 0) = 0\%$ and flux $c(0, y, t) = 100$ m^{-2}))

do not have necessarily multiple nodes at a same x coordinate (for example compare meshes in Figure 2.3(c) and (f)). It results that for unstructured meshes, more points are represented and the approximation seems to be better, which is of course deceptive[2] as all points are not located at the same y coordinate (and thus, no interpolations between all points can be made).

[2]Note that this 1D representation has no other purposes than enabling a comparison with an analytical solution, which has only been derived in 1D.

Other general observations can be made: refining the meshes leads to better results and maximal errors are observed in zones of high gradient of corrosion rate, as previously discussed.

2.3.2 Test Case 2

The test case 2 consist in a plate of dimensions 1×0.25 m subjected at its left edge, to a flux of 25 m^{-2} and to a constant corrosion rate of 25% at the right edge. The results shown in Figure 2.7 are very close to the analytical solution even for a coarse mesh. Results similar to those analyzed in 1D have indeed been recovered, which shows a proper behavior of the code. As each observations carried out in section 1.3.3 are still applicable, we invite the reader to report to this section for further discussions.

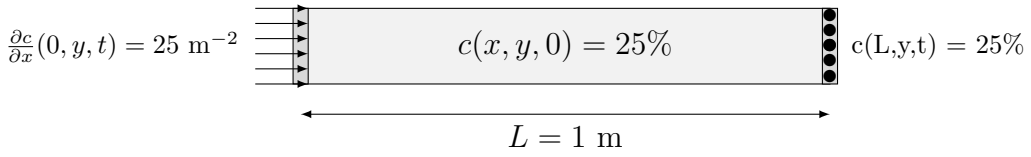


Figure 2.6: Initial and boundary conditions for study case 2

2.3.3 Conclusion

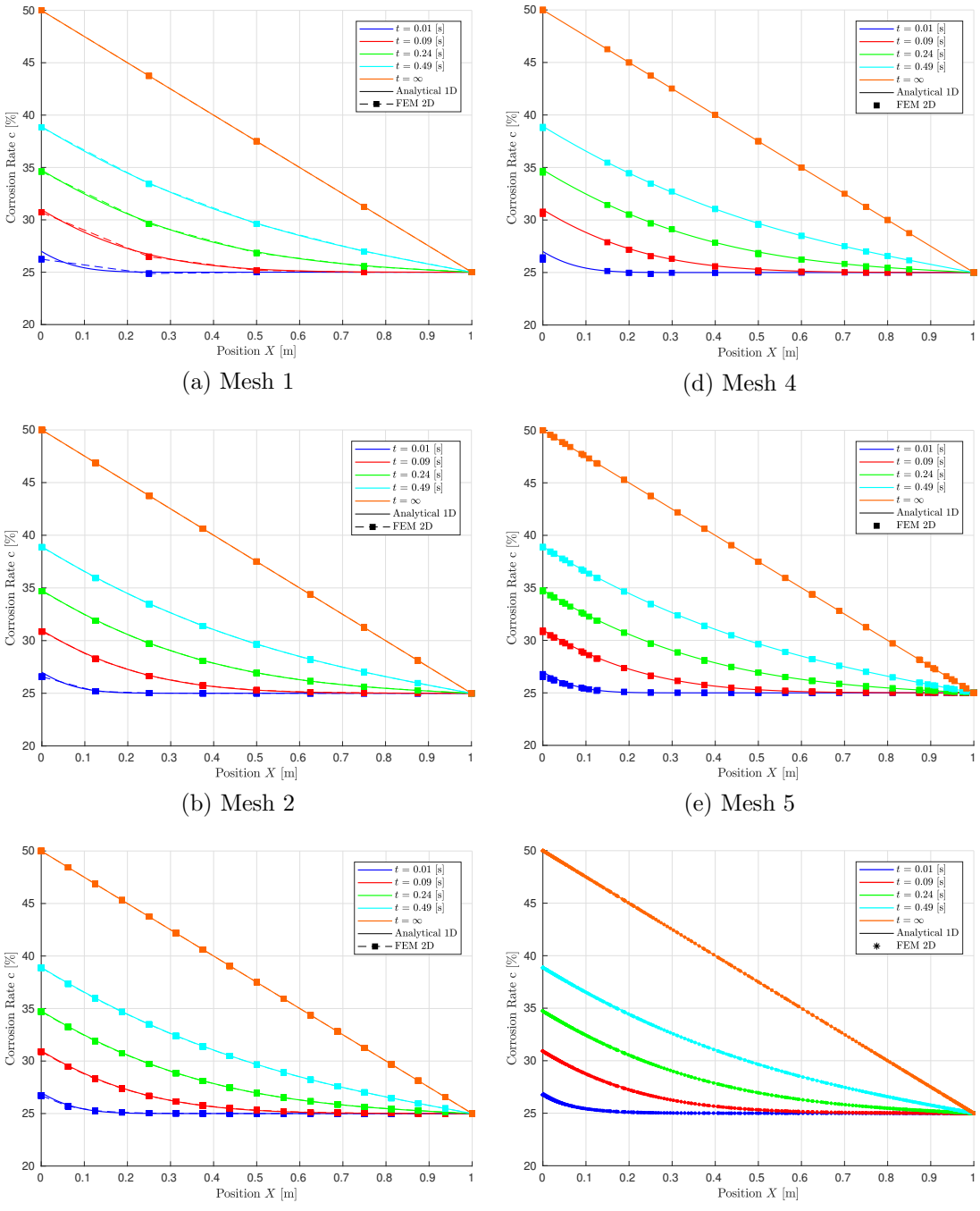
Refining the meshes leads indeed to better results and maximal errors are observed in zones of high gradient of corrosion rate, as previously discussed. The code accommodates unstructured meshes as well as structured ones. Both natural and essential boundary conditions have correctly been implemented as provided results match analytical solutions for both of them.

Note that the plots presented in this section are in 1D form for convenience purposes, in order to ease the comparison with analytical results. In the next section of this report, the results for corrosion will be presented in the form of a *colormap*, with a linear interpolation between values at nodes.

2.4 Validation of 2D model

Because no experimental results have been found in the literature[3], a proper validation has not been performed for this two dimensional model. The assumption that the diffusive behavior along the first dimension can be translated to the second dimension will therefore be relied on. This hypothesis is consistent with the assumed isotropic property of the body.

[3]Recall from section 1.4 that the experiments were modeling 1D diffusion as the literature study cases consisted in samples uniformly immersed in a furnace.



(a) Mesh 1

(d) Mesh 4

(b) Mesh 2

(e) Mesh 5

(c) Mesh 3

(f) Mesh 6

Figure 2.7: Results obtained from 2D-diffusion code, superimposed indifferently of their y position for each 6 meshes considered in test case 2 (parameters: $\Delta t = 0.01$ s, $D = 0.5$ m^2 s^{-1}, $c(L, y, t) = 25\%$, $c(x, y, 0) = 25\%$ and flux $\phi_c(0, y, t) = 25$ m^{-2}).

This page is intentionally left blank.

Chapter 3

2D Mechanical Model

In this chapter, a code to predict the behavior of the body under constraints will be implemented. These constraints can take the form of nodal or line loads, pressures, thermal strains, ... There exist of course many private and opensource codes/softwares that are able to compute finite elements and already provide very convenient ways to study the mechanical response of a solid to given constraints. A legitimate question that can arise is why then develop another code. In fact, the development described in this section have no other objective than facilitate the coupling between mechanical behavior and corrosion diffusion. Once verified and validated, the code will be particularized in the next chapters and its integration with the corrosion will then be eased.

3.1 Differential equation and body equilibrium

Starting from the general equations of equilibrium in a elementary volume in 3D,

$$\begin{cases} \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + \dfrac{\partial \tau_{xz}}{\partial z} + b_x = \rho a_x \\ \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + \dfrac{\partial \tau_{yz}}{\partial z} + b_y = \rho a_y \\ \dfrac{\partial \sigma_{zz}}{\partial z} + \dfrac{\partial \tau_{xz}}{\partial y} + \dfrac{\partial \tau_{yz}}{\partial z} + b_z = \rho a_z \end{cases} \tag{3.1}$$

Where b_i and a_i are respectively the acceleration and volumic load directed along direction i and ρ the body density. Assuming no dynamic and a plane stress state in xy, $a_x = a_y = a_z = 0$ and $\tau_{yz} = \tau_{xy} = \sigma_{zz} = 0$ are successively obtained. The system (3.1) becomes

$$\begin{cases} \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + b_x = 0 \\ \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + b_y = 0 \\ b_z = 0 \end{cases} \tag{3.2}$$

The last equation is trivial so it will not be considered in the further development; it will just be imposed to ensure that no force is acting out of plane. Finally, the

equations of static in plane stress state [26] are reduced to the following system

$$\begin{cases} \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + b_x = 0 \\ \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + b_y = 0 \end{cases} \tag{3.3}$$

3.2 Application to F.E.M.

Equation (3.3) requires the determination of the 2D displacement to be solved (the stress tensor components will be general functions of displacements). In other words, the vector field $\mathbf{u}(x, y) = (u(x, y),\ v(x, y))$ will be searched for such that (3.3) is satisfied. Now 2 unknowns are present in each point of the studied space, instead of a single unknown as in the previous chapters.

3.2.1 Constitutive law

The equation of continuity provides the expression of the strain vector $\boldsymbol{\varepsilon}$, which contains the 3 components of the deformation tensor ε_{ij}:

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

In plane stress state, only 3 components of ε_{ij} strain tensor are non zero. Rewritten on a matricial form, the non trivial components can be expressed as

$$\boldsymbol{\varepsilon} = \begin{Bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \gamma_{xy} \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial u}{\partial x} \\ \dfrac{\partial v}{\partial y} \\ \dfrac{\partial v}{\partial x} + \dfrac{\partial u}{\partial y} \end{Bmatrix} = \underbrace{\begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix}}_{=\mathbf{S}} \begin{Bmatrix} u \\ v \end{Bmatrix} = \mathbf{Su} \tag{3.4}$$

The matrix $\mathbf{S}$ is called the strain operator.

Provided that (3.3) expresses an equilibrium in stresses and applied loads, and that the system has to be solved for the displacements $u(x, y)$ and $v(x, y)$, it is essential to establish relationships between the displacement and stress fields. Such a relationship is given by the constitutive equation. In the scope of this master thesis[1], Hooke's law will be preferred as it is widely used in classical mechanics of solids, and the behavior of the studied body will be assumed to be elastic and linear. Furthermore, an isotropic behavior will be assumed.

[1]Remember that, as discussed in the introduction, the global purpose of this corrosion-mechanics coupling is to integrate the model developed in the `Solar Perform` project [4]. We start here to develop the mechanical behavior on its simplest form: linear elastic and isotropic.

With $\boldsymbol{\varepsilon}$ known by (3.4), the constitutive law can be used to derived the corresponding stress vector

$$\boldsymbol{\sigma} = \begin{Bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \tau_{xy} \end{Bmatrix} = \mathbf{D}\left(\boldsymbol{\varepsilon} - \boldsymbol{\varepsilon}_0\right) + \boldsymbol{\sigma}_0 \tag{3.5}$$

Where $\boldsymbol{\varepsilon}_0$ accounts for initial strains such as shrinkage or initial temperature (ΔT_0) and $\boldsymbol{\sigma}_0$ accounts for initial residual stress (confinement, etc, ...). The matrix $\mathbf{D}$ can be particularized depending on the constitutive law used. In the present case (Hooke's law), the matrix D takes the form

$$\mathbf{D} = \frac{E}{1-\nu^2}\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix} \tag{3.6}$$

3.2.2 Derivation of the weak form of the problem

As suggested by [27], the problem defined by equation (3.3) and associated with suitable boundary condition can be rewritten under the form of two differential operators $\mathbf{A}(\mathbf{u})$ and $\mathbf{B}(\mathbf{u})$:

$$\mathbf{A}(\mathbf{u}) = \mathbf{0} \qquad \text{in V} \tag{3.7}$$

$$\mathbf{B}(\mathbf{u}) = \mathbf{M}\mathbf{u} + \mathbf{t} = \mathbf{0} \qquad \text{in } \Gamma \tag{3.8}$$

By identification

$$\mathbf{A}\left(\mathbf{u}\right) = \mathbf{0} \quad \Leftrightarrow \quad \begin{Bmatrix} A_1 \\ A_2 \end{Bmatrix} = \begin{Bmatrix} \dfrac{\partial \sigma_{xx}}{\partial x} + \dfrac{\partial \tau_{xy}}{\partial y} + b_x \\ \dfrac{\partial \sigma_{yy}}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + b_y \end{Bmatrix} = \mathbf{0} \tag{3.9}$$

This set of equations has to be zero in each point of the domain V. It follows that by multiplying each equation by respectively δu and δv, and summing the two resulting equations, we can write:

$$A_1 \delta u + A_2 \delta v = 0 \tag{3.10}$$

As an approximation function $\tilde{\mathbf{u}} \approx \mathbf{u}$ that verifies (3.10) on average over a chosen domain V is needed, it follows that:

$$\mathtt{WR} = \int_V A_1 \delta u + A_2 \delta v \; dV = 0 \tag{3.11}$$

Or, by (3.9)

$$\mathtt{WR} = \int_V \left(\frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} + b_x \right) \delta u + \left(\frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{xy}}{\partial x} + b_y \right) \delta v \; dV = 0 \tag{3.12}$$

Using Green's theorem, we get

$$\mathtt{WR} = \oint_\Gamma (\sigma_{xy}\delta u + \tau_{xy}\delta v)\cdot n_x + (\tau_{xy}\delta u + \sigma_{yy}\delta v)\cdot n_y \, d\Gamma - \int_V \sigma_{xx}\frac{\partial \delta u}{\partial x} + \tau_{xy}\frac{\partial \delta v}{\partial x} + \sigma_{yy}\frac{\partial \delta v}{\partial y} + \tau_{xy}\frac{\partial \delta u}{\partial y} + b_x\delta u + b_y\delta v \, dV = 0 \tag{3.13}$$

In the previous equation, three dot products where the vectors δu and $\boldsymbol{\sigma}$ appear can be identified, and

$$\delta\boldsymbol{\varepsilon}^T = \begin{bmatrix} \dfrac{\partial \delta u}{\partial x} & \dfrac{\partial \delta v}{\partial y} & \dfrac{\partial \delta v}{\partial x} + \dfrac{\partial \delta u}{\partial y} \end{bmatrix}$$

The principle of virtual work statement is then recovered:

$$\mathtt{WR} = \int_V \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} \, dV - \int_V \delta\mathbf{u}^T\mathbf{b} \, dV - \oint_\Gamma \delta\mathbf{u}^T\mathbf{t} \, d\Gamma = 0 \tag{3.14}$$

The formulation proposed by (3.13) and (3.14) are then equivalent, indicating that the weak form has been properly derived. The line integral represents the effects of boundary conditions and will be left for discussion to further section 3.2.7. At this stage, this term is simply neglected without more justification. Taking into consideration this last remark, the weak form of the problem is given by the two equivalent formulations

$$\int_V \left(\sigma_x\frac{\partial \delta u}{\partial x} + \tau_{xy}\frac{\partial \delta v}{\partial x} + \sigma_{yy}\frac{\partial \delta v}{\partial y} + \tau_{xy}\frac{\partial \delta u}{\partial y} + b_x\delta u + b_y\delta v \right) \, dV = 0 \tag{3.15}$$

$$\int_V \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma} \, dV - \int_V \delta\mathbf{u}^T\mathbf{b} \, dV = 0 \tag{3.16}$$

3.2.3 Expression of stiffness matrix and nodal load vector

To keep it as general as possible, it will be assumed that the types of element considered are n_{nod} nodes elements. The mathematical developments presented in this section are then valid whatever the type of considered element (provided that they are 2D elements). First of all, the displacement $\mathbf{u}$ is interpolated as

$$\mathbf{u}(x,y) = \begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} \approx \tilde{\mathbf{u}}(x,y) = \begin{Bmatrix} \tilde{u}(x,y) \\ \tilde{v}(x,y) \end{Bmatrix} = \mathbf{N}\mathbf{q} \tag{3.17}$$

With $\mathbf{q}$ the nodal unknown vector that takes the form (n_{nod} still refers to the number of node by element)

$$\begin{Bmatrix} \tilde{u} \\ \tilde{v} \end{Bmatrix} = \begin{bmatrix} N_1(\xi,\eta) & 0 & N_2(\xi,\eta) & 0 & \cdots & N_{n_{nod}}(\xi,\eta) & 0 \\ 0 & N_1(\xi,\eta) & 0 & N_2(\xi,\eta) & \cdots & 0 & N_{n_{nod}}(\xi,\eta) \end{bmatrix} \begin{Bmatrix} q_{u,1} \\ q_{v,1} \\ q_{u,2} \\ q_{v,2} \\ \vdots \\ q_{u,n_{nod}} \\ q_{v,n_{nod}} \end{Bmatrix} \tag{3.18}$$

Alternatively, the interpolated displacement can be expressed as

$$\begin{aligned} u(x,y) &= \sum_{i=1}^{n_{nod}} N_{1i}(x,y) \cdot q_{2i-1} \\ v(x,y) &= \sum_{i=1}^{n_{nod}} N_{2i}(x,y) \cdot q_{2i} \end{aligned} \tag{3.19}$$

This interpolation is valid everywhere inside the considered finite element. This time, $\mathbf{N}$ is not a vector, but a $2 \times n_{dof}$ matrix. The strain $\boldsymbol{\varepsilon}$ defined at (3.4), can also be calculated as a function of the nodal unknowns $\mathbf{q}$:

$$\boldsymbol{\varepsilon} = \mathbf{S}\mathbf{u} = \mathbf{S}\left(\mathbf{N}\mathbf{q}\right) = \mathbf{B}\mathbf{q} \tag{3.20}$$

Where $\mathbf{B} = \mathbf{S}\mathbf{N}$ is the strain shape function matrix. The same operation is repeated with $\boldsymbol{\sigma}$: from (3.17) and the simplest form of (3.5), we get

$$\boldsymbol{\sigma} = \mathbf{D}\boldsymbol{\varepsilon} = \mathbf{D}\mathbf{B}\mathbf{q} \tag{3.21}$$

From equations (3.17) and (3.20), it comes directly that

$$\begin{aligned} \mathbf{u} = \mathbf{N}\mathbf{q} \quad &\Rightarrow \quad \delta\mathbf{u} = \mathbf{N}\delta\mathbf{q} \\ \boldsymbol{\varepsilon} = \mathbf{B}\mathbf{q} \quad &\Rightarrow \quad \delta\boldsymbol{\varepsilon} = \mathbf{B}\delta\mathbf{q} \end{aligned}$$

Substituting all these expressions into weak form (3.16),

$$\mathtt{WR} = \int_V \delta\mathbf{q}^T\mathbf{B}^T\mathbf{D}\mathbf{B}\mathbf{q}\, dV - \int_V \delta\mathbf{q}^T\mathbf{N}^T\mathbf{b}\, dV = 0 \tag{3.22}$$

$$\Leftrightarrow \mathtt{WR} = \delta\mathbf{q}^T e\left(\int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\, d\Omega\, \mathbf{q} - \int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega\right) = 0 \tag{3.23}$$

Because $\delta\mathbf{q}$ are arbitrary coefficients, the last equation degenerates in

$$\int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\, d\Omega\, \mathbf{q} = \int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega \tag{3.24}$$

Which is a linear system

$$\mathbf{K}_e\mathbf{q} = \mathbf{g}_e$$

The expressions of local stiffness matrix and nodal vector are then derived:

$$\mathbf{K}_e = \int_\Omega \mathbf{B}^T\mathbf{D}\mathbf{B}\, d\Omega \tag{3.25}$$

$$\mathbf{g}_e = \int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega \tag{3.26}$$

Note that these expressions (3.25) and (3.26) are valid whatever the type of elements and whatever the constitutive law.

3.2.4 Global stiffness matrix and nodal load vector

As previously explained, knowing the stiffness matrix for each element of the domain, the global stiffness matrix can be built performing an assembly operation:

$$\mathbf{K} = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \mathbf{K}_e = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \int_\Omega \mathbf{B}^T \mathbf{D} \mathbf{B} \, dx \, dy \tag{3.27}$$

$$\mathbf{g} = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \mathbf{g}_e = \underset{el=1}{\overset{N_{el}}{\mathrm{A}}} \int_\Omega \mathbf{N}^T \mathbf{b} \, dx \, dy \tag{3.28}$$

Solving the linear system $\mathbf{Kq} = \mathbf{g}$ for $\mathbf{q}$ provides the approximated displacement $\tilde{\mathbf{u}}$ at each node of the mesh.

3.2.5 Shape functions

The weak form (3.16) involves at most the first spatial derivative of displacement field $\mathbf{u}$. If linear shape functions are used, the displacements are then linear functions in ξ and η while strains $\boldsymbol{\varepsilon}$ are linear either in ξ or in η. It means that linear shape functions would correctly (i.e. not trivially) represent each term of (3.16). Linear shape functions (i.e. Quad4 mapped element) seem then to be an appropriate choice.

Because section 2.2.4 already details the shape functions, the same information will not be presented, and only the main points as well as what changes from previous chapters will be mentionned.

$$N_1(\xi,\eta) = \frac{1}{4}(1-\xi)(1-\eta)$$

$$N_2(\xi,\eta) = \frac{1}{4}(1-\xi)(1+\eta)$$

$$N_3(\xi,\eta) = \frac{1}{4}(1+\xi)(1+\eta)$$

$$N_4(\xi,\eta) = \frac{1}{4}(1+\xi)(1-\eta)$$

Each shape function is associated with one node of the element and are represented in Figure 2.2. Because 2 degrees of freedom are associated with one node, the shape function matrix contains as many zeros as shape functions:

$$\mathbf{N} = \begin{bmatrix} N_1(\xi,\eta) & 0 & N_2(\xi,\eta) & 0 & N_3(\xi,\eta) & 0 & N_4(\xi,\eta) & 0 \\ 0 & N_1(\xi,\eta) & 0 & N_2(\xi,\eta) & 0 & N_3(\xi,\eta) & 0 & N_4(\xi,\eta) \end{bmatrix} \tag{3.29}$$

The strain shape function matrix $\mathbf{B}$ is

$$\mathbf{B} = \mathbf{SN} = \begin{bmatrix} \dfrac{\partial}{\partial x} & 0 \\ 0 & \dfrac{\partial}{\partial y} \\ \dfrac{\partial}{\partial y} & \dfrac{\partial}{\partial x} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}$$

And then,

$$\mathbf{B}(\xi,\eta) = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & 0 & \dfrac{\partial N_2}{\partial x} & 0 & \dfrac{\partial N_3}{\partial x} & 0 & \dfrac{\partial N_4}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1}{\partial y} & 0 & \dfrac{\partial N_2}{\partial y} & 0 & \dfrac{\partial N_3}{\partial y} & 0 & \dfrac{\partial N_4}{\partial y} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & \dfrac{\partial N_2}{\partial y} & \dfrac{\partial N_2}{\partial x} & \dfrac{\partial N_3}{\partial y} & \dfrac{\partial N_3}{\partial x} & \dfrac{\partial N_4}{\partial y} & \dfrac{\partial N_4}{\partial x} \end{bmatrix} \tag{3.30}$$

The components of $\mathbf{B}$ may be determined indirectly using the jacobian of the regular change of variables:

$$\frac{\partial N_i}{\partial x} = \frac{\partial N_i}{\partial \xi} \cdot \frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta} \cdot \frac{\partial \eta}{\partial x}$$

$$\frac{\partial N_i}{\partial y} = \frac{\partial N_i}{\partial \xi} \cdot \frac{\partial \xi}{\partial y} + \frac{\partial N_i}{\partial \eta} \cdot \frac{\partial \eta}{\partial y}$$

With $\frac{\partial \xi}{\partial x}$, $\frac{\partial \eta}{\partial x}$, $\frac{\partial \xi}{\partial y}$ and $\frac{\partial \eta}{\partial y}$ that can be determined by identification inverting the jacobian matrix $\mathbf{J}$

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \xi} & \dfrac{\partial x}{\partial \eta} \\ \dfrac{\partial y}{\partial \xi} & \dfrac{\partial y}{\partial \eta} \end{bmatrix} \qquad \longrightarrow \qquad \mathbf{J}^{-1} = \begin{bmatrix} \dfrac{\partial \xi}{\partial x} & \dfrac{\partial \xi}{\partial y} \\ \dfrac{\partial \eta}{\partial x} & \dfrac{\partial \eta}{\partial y} \end{bmatrix}$$

The integration domain of (3.25) and (3.26) must be modified in accordance to the change of variable $(x, y) \to (\xi, \eta)$. Furthermore, the jacobian of the regular transformation is also introduced. Then, the expressions of stiffness matrix and nodal load vector becomes

$$\mathbf{K}_e = \int_{-1}^{1}\int_{-1}^{1} \mathbf{B}^T\mathbf{D}\mathbf{B} \cdot J \; d\xi \; d\eta \tag{3.31}$$

$$\mathbf{g}_e = \int_{-1}^{1}\int_{-1}^{1} \mathbf{N}^T\mathbf{b} \cdot J \; d\xi \; d\eta \tag{3.32}$$

3.2.6 Numerical integration

Equation (3.31) and (3.32) both require the evaluation of a surface integral. Since the principle has already been discussed in detail in section 2.2.5, and equations (3.31) and (3.32) do not rise other issues than the ones addressed previously, the reader will be referred to the aforementionned section for further information.

3.2.7 Boundary conditions

The line integral in (3.13) has been left aside in the definition of the stiffness matrix and nodal load vector. This integral incorporates the effects of the boundary condition to the linear system. For essential boundary conditions, the integral is non zero only if a non zero displacement is imposed at one considered node.

Nodal loads are applied on the structure directly at the adequate line of the nodal load vector, while line loads require an equivalent nodal loads vector to be calculated (and from that point, line loads are treated identically to common nodal loads).

Note that the stiffness matrix as detailed by (3.27) is singular; the system is consequently badly conditioned. It is therefore expected to enforce several boundary conditions[2] in order to avoid any rigid body mode such as translation and rotation. At least two degrees of freedom must therefore be blocked in both u and v direction.

3.2.8 Calculation of strains/stresses and local smoothing

Once the displacements $\mathbf{u}$ are known, the strains are accordingly calculated at gauss points as suggested by (3.20)

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{Bq}$$

And the stresses $\hat{\boldsymbol{\sigma}}$ at gauss points

$$\hat{\boldsymbol{\sigma}} = \mathbf{DBq}$$

Many authors [28] have shown that for displacements based finite element approach, stresses sampled at nodal points are usually very poor representations of the real stress values: for any method using $\mathbb{C}_0$ elements, inter-element continuity is respected for the displacement field only while first derivatives and thus, strains and stress fields, experience unrealistic "jumps". As a result, stresses should never be directly evaluated at nodes. The most relevant locations for the sampled points are Gauss points. For practical considerations inherent to the post-processing software, it is however necessary to calculate stresses sampled at nodes rather than at gauss points. A stress recovering procedure is then required.

Numerous authors [29–31] have proposed various stress recovery procedures for displacement based approach. [32,33] have based their simple approaches on least square method, performing a stress extrapolation as well as a local smoothing.

Suppose that $\tilde{\boldsymbol{\sigma}}$ refers to the stresses sampled at nodes (i.e. the researched vector) and $\hat{\boldsymbol{\sigma}}$ the stress sampled at gauss points. Assuming that an approximation of the real stresses can be interpolated by a polynomial, we get

$$\boldsymbol{\sigma}(\xi,\eta) \approx \boldsymbol{\sigma}^*(\xi,\eta) = \mathbf{N}_{\boldsymbol{\sigma}}(\xi,\eta)\tilde{\boldsymbol{\sigma}} \tag{3.33}$$

In other words, it is assumed that the approximation of stresses inside a finite element is a polynomial function whose coefficients are the value of stress at nodes. [28, 34] suggested an obvious choice for $\mathbf{N}_{\boldsymbol{\sigma}}$ such that $\mathbf{N}_{\boldsymbol{\sigma}} = \mathbf{N}$. The averaged error on node in sense of least square method is given by the functional

$$\Pi = \int_\Omega \left(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*\right)^2 \, d\Omega = \int_\Omega \left(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*\right)^T \left(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*\right) \, d\Omega \tag{3.34}$$

[2]Imposing one boundary condition reduces by 1 the rank of the stiffness matrix, and thus removes one linear dependence of the global stiffness matrix. Mathematically, by imposing a sufficient amount of boundary conditions, one intends removing all linear dependencies between the lines.

The exact solution $\boldsymbol{\sigma}$ is recovered if $\Pi = 0$, but because $\boldsymbol{\sigma}$ was assumed to be interpolated by a finite set of polynomials $\mathbf{N}_{\boldsymbol{\sigma}}$, this result will never be observed. The optimal solution corresponding to the generalized displacements $\mathbf{q}$ is such that the function Π is minimal:

$$\min\left[\Pi\right] = \min\left[\int_{\Omega} \left(\boldsymbol{\sigma} - \boldsymbol{\sigma}^{*}\right)^{2} \, d\Omega\right] \tag{3.35}$$

Substituting (3.33), we get

$$\min\left[\int_{\Omega} \left(\boldsymbol{\sigma} - \mathbf{N}_{\boldsymbol{\sigma}}\tilde{\boldsymbol{\sigma}}\right)^{2} \, d\Omega\right] \tag{3.36}$$

The solution is found enforcing

$$\frac{\partial \Pi}{\partial \tilde{\boldsymbol{\sigma}}} = 0 \Leftrightarrow \int_{\Omega} \sigma \mathbf{N}_{\boldsymbol{\sigma}} - \mathbf{N}_{\boldsymbol{\sigma}}^{T}\mathbf{N}_{\boldsymbol{\sigma}}\tilde{\boldsymbol{\sigma}} \, d\Omega = 0 \tag{3.37}$$

This degenerate in a linear system to solve for $\tilde{\boldsymbol{\sigma}}$

$$\int_{\Omega} \mathbf{N}_{\boldsymbol{\sigma}}^{T}\mathbf{N}_{\boldsymbol{\sigma}} \, d\Omega \, \tilde{\boldsymbol{\sigma}} = \int_{\Omega} \boldsymbol{\sigma}\mathbf{N}_{\boldsymbol{\sigma}} \, d\Omega \tag{3.38}$$

Using dimensionless coordinate system,

$$\int_{-1}^{1}\int_{-1}^{1} \mathbf{N}_{\boldsymbol{\sigma}}^{T}\mathbf{N}_{\boldsymbol{\sigma}} \cdot J \, d\xi \, d\eta \, \tilde{\boldsymbol{\sigma}} = \int_{-1}^{1}\int_{-1}^{1} \boldsymbol{\sigma}\mathbf{N}_{\boldsymbol{\sigma}} \cdot J \, d\xi \, d\eta \tag{3.39}$$

The last equation (3.39) requires at first glance to know the expression of $\boldsymbol{\sigma}$ (which is naturally never achieved). However, if integrals are numerically evaluated through a sum over weighted integrants evaluated at gauss points (Gauss Legendre quadrature), it is no longer required to know $\boldsymbol{\sigma}$ to solve the (3.39):

$$\left(\sum_{i=1}^{n}\sum_{j=1}^{n} \mathbf{N}_{\boldsymbol{\sigma}}^{T}(\xi_i, \xi_j)\mathbf{N}_{\boldsymbol{\sigma}}(\xi_i, \xi_j) \cdot w_i w_j \cdot J\right) \tilde{\boldsymbol{\sigma}} = \sum_{i=1}^{n}\sum_{j=1}^{n} \hat{\boldsymbol{\sigma}}(\xi_i, \xi_j)\mathbf{N}_{\boldsymbol{\sigma}}(\xi_i, \xi_j) \cdot w_i w_j \cdot J \tag{3.40}$$

This last equation enables a local stress smoothing and extrapolation from gauss points to nodes, in the sense of least squared.

3.3 Model verification

The model verification will be conducted on two test cases, on both regular and irregular meshes. A patch test will also be performed for the first test case, as the second one involves non constant strain. Note that because the chosen elements satisfy all the continuity requirements and are exactly integrated (reduced integration was not used), this patch test may seem unnecessary [35]. Factually, it makes sense as it gives an efficient way to check that the code has been properly implemented.

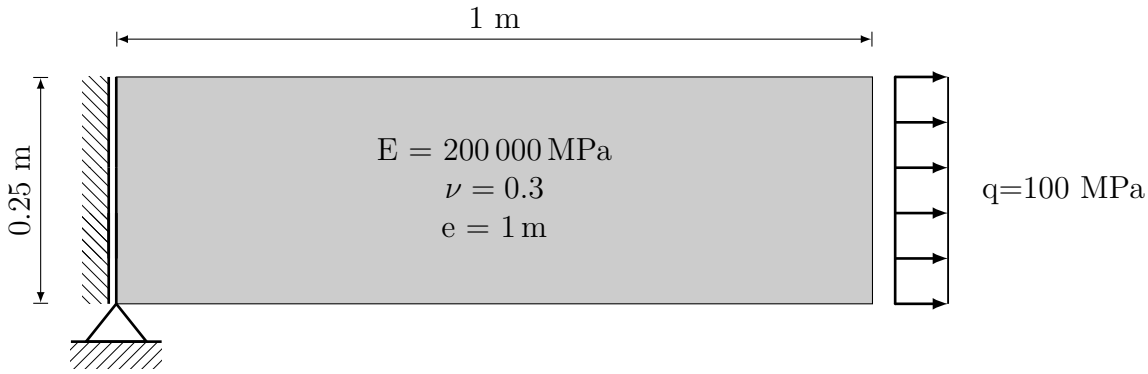
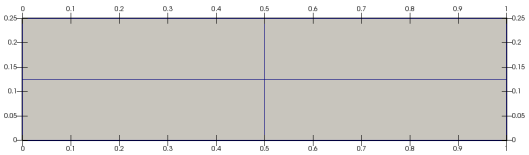
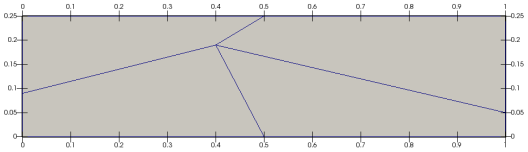


Figure 3.1: Description of test case 1.

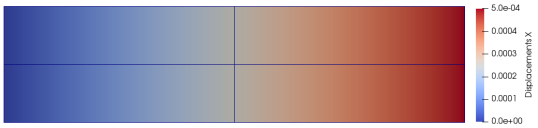


(a) Mesh 1 (16 nodes, 4 elements)

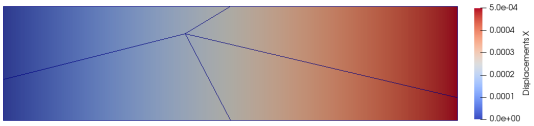


(b) Mesh 2 (16 nodes, 4 elements), used for patch test

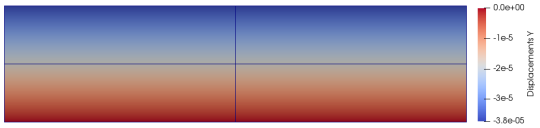
Figure 3.2: Used meshes for the first test case



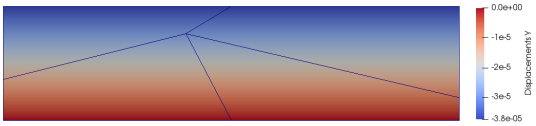
(a) Displacements u [m]



(f) Displacements u [m]



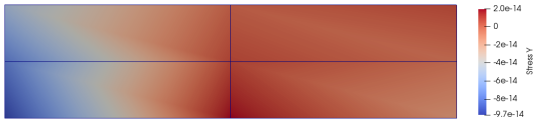
(b) Displacements v [m]



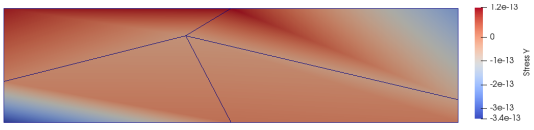
(g) Displacements v [m]

(c) Stresses σ_{xx} [MPa]

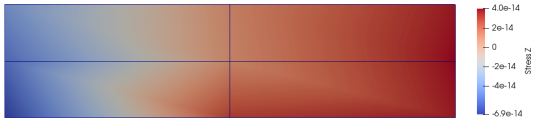
(h) Stresses σ_{xx} [MPa]



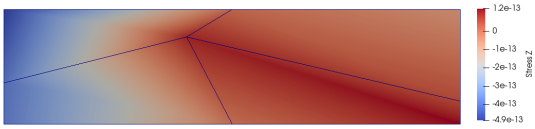
(d) Stresses σ_{yy} [MPa]



(i) Stresses σ_{yy} [MPa]



(e) Stresses τ_{xz} [MPa]



(j) Stresses τ_{xy} [MPa]

Figure 3.3: Results for test case 1

3.3.1 Test Case 1

The first test case consists in a plate in extension: a schematic drawing of the situation is shown at Figure 3.1. The considered meshes may be seen in Figure 3.2.

For such basic loading, the analytical solution can be expressed from Hooke's law:

$$\varepsilon_{ij} = \frac{1}{E}\left[(1+\nu)\sigma_{ij} - \nu\sigma_{kk}\delta_{ij}\right]$$

Because $\sigma_{11} = 0.25f/A$ and $\sigma_{22} = 0$,

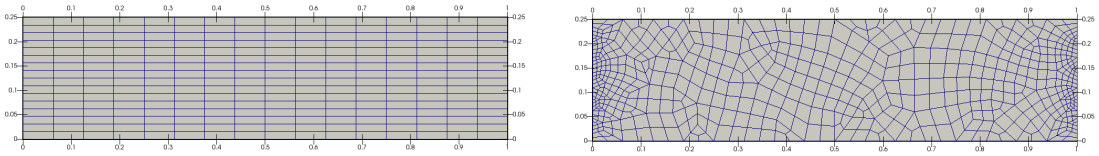
$$\varepsilon_{11} = \frac{0.25f}{EA} \qquad \rightarrow u(x) = \frac{0.25f}{EA}\cdot x$$

$$\varepsilon_{22} = -\nu\frac{0.25f}{EA} \qquad \rightarrow v(y) = -\nu\frac{0.25f}{EA}\cdot y$$

Substituting $f = 100\text{kN}\,\text{m}^{-1} \times 0.25\text{m}$, $E = 200000\text{MPa}$ and $\nu = 0.3$, the maximal horizontal displacement at $x = L$ is 5×10^{-4} m and the maximal vertical displacement at $y = 0.25\text{m}$ is 3.7510^{-5}m. The stresses $\sigma_{11} = f \cdot 0.25/(0.25 \cdot 1) = 100\text{kPa}$, $\sigma_{22} = \tau_{12} = 0$. These values are accurately recovered by the numerical model, as can be observed in Figure 3.3. It can thus be concluded that the first test case is correctly tackled by the code.

3.3.2 Test Case 2

The second test case is about in-plane plate bending. The considered situation is shown in Figure 3.5. Here again, the situation will be assessed on two meshes. A patch test may be carried out, since in-plane bending involves gradients of stress that can only be correctly modeled if the mesh is sufficiently fine.



(a) Mesh 1 (289 points, 256 cells) (b) Mesh 2 (663 points, 598 cells)

Figure 3.4: Meshes used for first test case of verification step

Results are shown in Figure 3.5. Slight differences ($< 2\%$) in displacements may be observed between results provided by structured and unstructured meshes, showing that unstructured meshes should be further refined to get close to the analytical solution.

Larger differences are however observed for the stress, even if the global stress pattern remains the same (e.g. the lower part in tension and upper part in compression for σ_{xx}). The shear stresses are parasitized by light "stress jumps". These irregularities have no physical sense and are due to the stress recovery procedure. They occur especially in zones where quadrangle elements are close to triangle elements. When the generated mesh is such that 3 points of a quadrangle

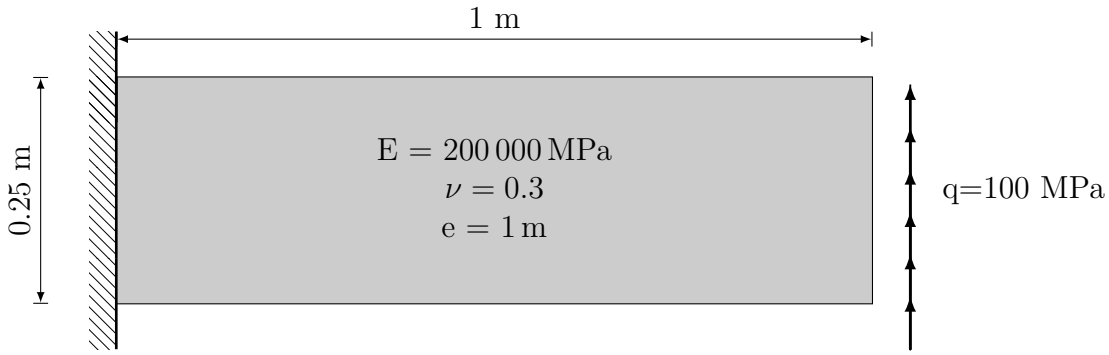
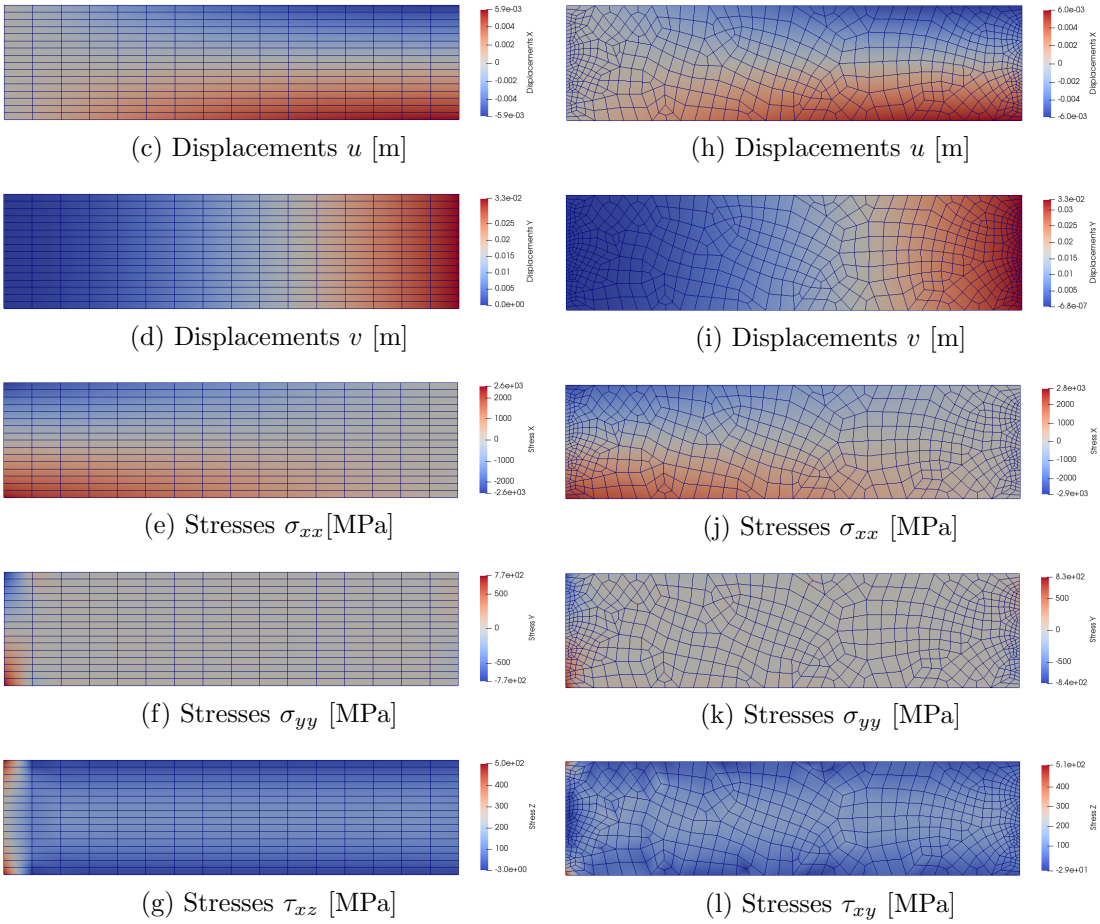


Figure 3.5: Description of test case 2.

are nearly aligned, the polynomial approximating the stress inside the element is not of a sufficiently high degree to properly model the stress and consequently generates inter-elements discontinuities. As these irregularities do not disturb the interpretation of the results (the global stress pattern is still correctly identifiable), they can be neglected. The unconvinced reader may refer to A2, where second order shape functions were used for the same application and where smoother stress distribution is visible in Figure A.2.



(c) Displacements u [m] (h) Displacements u [m]

(d) Displacements v [m] (i) Displacements v [m]

(e) Stresses σ_{xx}[MPa] (j) Stresses σ_{xx} [MPa]

(f) Stresses σ_{yy} [MPa] (k) Stresses σ_{yy} [MPa]

(g) Stresses τ_{xz} [MPa] (l) Stresses τ_{xy} [MPa]

Figure 3.5: Results for test case 2

To conclude this section, it will be emphasized that the numerical developments accommodate regular as well as irregular meshes. Numerical developments have been derived on a displacement based approach, so it is always expected to get

higher errors on the stress fields than on the displacements field. The accuracy of the predictions of the stress fields are entirely dependent on the stress extrapolation procedure, which aims to minimize the error on stress inside an element. As a result, it should be kept in mind that high discontinuities in stress field, if they exist in the real application, will be smoothed by the stress recovery process.

The presented results in Figure 3.5 for second test case have been compared with a reference solution. This solution was obtained by simulation with the finite elements software `SAP2000`, for same hypothesis. Results are not presented in the core of the text, but may be consulted in Appendix A in Figure A.3.

3.4 Model validation

It is commonly admitted that the system (3.1) describes exactly the behavior of deformable solids whether or not these experience elastic behavior. The use of Hooke's law (3.5) restricts however the use of the model for elastic behavior only. Nevertheless, a nonlinear behavior can be simulated by adopting a variable Young modulus E (this choice of a varying E will be discussed in the next section).

In accordance with those considerations, the code is validated keeping in mind the restricting assumptions:

- The code neither accounts for plastic deformations nor for second order effects
- The stress state is assumed to be plane.
- The behavior is punctually isotropic, but the code can accommodate non-homogeneous properties distributed over the plate. It is clear that this assumption becomes invalid if the studied domain involves a highly irregular distribution of material properties (in which case full anisotropy should be considered).

3.5 Conclusion

The objective of this chapter was to develop numerical model for the mechanical behavior of studied solids, in order to facilitate the development of the coupling model.

The code has been verified based on two test cases, with analytical solution or a reference solution obtained by a trusted finite element analysis software. In all cases, the developed numerical model matched the reference solutions for both structured and unstructured meshes.

Few irregularities have been found in the stress predictions. This is due to the fact that the developed procedure relies on a displacement based approach: the nodal unknowns are the displacement field u and v, but not the stresses. To predict the stresses (that are only known at integration points), it was necessary

to develop a stress extrapolation procedure. The one considered consists in the determination of stresses at nodes in order to minimize the averaged error inside the element (in the least square sense). As a result, a smoothing (local smoothing) of the stress distribution may occur if gradients of stresses are high.

Based on this consideration, the mechanical model provides entire satisfaction and allows us to proceed with the coupling of corrosion and mechanical model.

Chapter 4

Coupling between corrosion and mechanics

Previous chapters have developed corrosion diffusion and mechanical behavior as independent phenomena. Now it is time to model them as interdependent, coupling them both in a single linear system, which shall accounts for the influence of the corrosion on the mechanical behavior as well as the influence of the stress state on the diffusive phenomenon.

If the corrosion diffusion was assumed to follow a *fickian diffusion*, we will now add one term accounting for a *non fickian flux* in the differential equation governing the diffusive phenomenon. Therefore, from now on, the diffusion phenomenon is no more purely diffusive. The equations governing the mechanical behavior remain, for their part, unchanged. However, the constitutive law will be modified in order to take corrosion induced damage on the material stiffness into account (even if a linear isotropic behavior will still be assumed).

4.1 Derivation of the weak forms

4.1.1 Corrosion aspects

As suggested by [2, 3, 36], the mass transport phenomenon may be studied based on equation

$$\frac{\partial c}{\partial t} = \nabla \cdot (D\nabla c - DMc\nabla p) \tag{4.1}$$

With p the hydrostatic pressure $p = \frac{1}{2}(\sigma_{11} + \sigma_{22})$ and M the pressure factor. The first term of the right hand side is purely diffusive, also called the fickian flux J_F while the second, stress dependent, is called a non fickian flux. Such diffusion process may be qualified as **stress driven diffusion**. The previous chapters have particularized this equation assuming $M = 0$. In this chapter, we will integrate the effect of this second term and hence, this equation will be considered as it stands.

The weighted residual may be formed multiplying (4.1) by the weighting function $\delta c(x, y)$ and integrating over the volume defined by an element of surface Ω

and thickness e,

$$\texttt{WR} = \int_V \frac{\partial c}{\partial t}\delta c \, dV - e\int_V \nabla\cdot(D\nabla c - DMc\nabla p)\,\delta c \, dV \tag{4.2}$$

$$\Leftrightarrow \texttt{WR} = e\int_\Omega \frac{\partial c}{\partial t}\delta c \, d\Omega - e\int_\Omega \nabla\cdot(D\nabla c - DMc\nabla p)\,\delta c \, d\Omega \tag{4.3}$$

An integration by parts combined with Gauss theorem gives

$$\begin{aligned}\texttt{WR} = e\int_\Omega \frac{\partial c}{\partial t}\delta c \, d\Omega - e\int_\Gamma (D\nabla c - DMc\nabla p)\,\delta c \, d\Gamma \\ + eD\int_\Omega (D\nabla c - DMc\nabla p)\cdot\nabla(\delta c)\, d\Omega = 0\end{aligned} \tag{4.4}$$

The line integral represents the influences of boundary conditions and is treated in accordance with sections 1.2.7 and 3.2.7: its value is zero if the considered element experiences do not exchange with the external world, but must be considered as a independent term (nodal load) for nodes subjected to incoming/outgoing fluxes. In order to ease the readability, we will pursuit the mathematical developments assuming that the body experiences no exchange with the exterior world (this does not change the expressions of stiffness matrices). Effect of boundary conditions will be modeled by adding an *ad-hoc* contribution to the nodal load vector (natural boundary condition), as explained in section 1.2.7, and/or modifying adequately the stiffness matrix (essential boundary condition).

Developing the expressions of gradient and assuming D constant everywhere inside the finite element, we get

$$\texttt{WR} = e\int_\Omega \frac{\partial c}{\partial t}\delta c \, d\Omega + eD\int_\Omega \frac{\partial c}{\partial x}\cdot\frac{\partial \delta c}{\partial x} + \frac{\partial c}{\partial y}\cdot\frac{\partial \delta c}{\partial y} - Mc\frac{\partial p}{\partial x}\cdot\frac{\partial \delta c}{\partial x} - Mc\frac{\partial p}{\partial y}\cdot\frac{\partial \delta c}{\partial y}\, d\Omega \tag{4.5}$$

Finally, the weak form statement is obtained enforcing the weighted residual WR to 0

$$e\int_\Omega \frac{\partial c}{\partial t}\delta c \, d\Omega + eD\int_\Omega \frac{\partial c}{\partial x}\cdot\frac{\partial \delta c}{\partial x} + \frac{\partial c}{\partial y}\cdot\frac{\partial \delta c}{\partial y} - Mc\frac{\partial p}{\partial x}\cdot\frac{\partial \delta c}{\partial x} - Mc\frac{\partial p}{\partial y}\cdot\frac{\partial \delta c}{\partial y}\, d\Omega = 0 \tag{4.6}$$

This last equation constitutes the weak form of the corrosion problem.

4.1.2 Mechanical aspects

Since there is nothing more to introduce in the mechanical model, we will consider the same weak form as the one obtained at (3.16).

$$\int_V \delta\boldsymbol{\varepsilon}^T\boldsymbol{\sigma}\, dV - \int_V \delta\mathbf{u}^T\mathbf{b}\, dV = 0 \tag{4.7}$$

4.2 Application to FEM

4.2.1 Coupling matrix

Up to now, we considered corrosion and mechanics as independent phenomena, using the weighted residual method to derive one stiffness matrix and one nodal

load vector for each of them. In a coupling model, the stiffness matrix enables to link both parts of the problem, and takes the form

$$\mathbf{Kq} = \mathbf{g} \quad \Leftrightarrow \quad \begin{bmatrix} \mathbf{K}_{cc} & \mathbf{K}_{cu} \\ \mathbf{K}_{uc} & \mathbf{K}_{uu} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_c \\ \mathbf{q}_u \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_c \\ \mathbf{g}_u \end{Bmatrix} \tag{4.8}$$

In this global stiffness matrix, the diagonal terms $\mathbf{K}_{cc}$ and $\mathbf{K}_{uu}$ are already known, since they have been derived in Chapter 2 and Chapter 3: the developments carried in this chapter should then lead to the same expressions than (3.31) and (2.25) for $\mathbf{K}_{cc}$ and $\mathbf{K}_{uu}$ respectively. The novelty tackled by this chapter are the anti-diagonal stiffness matrices $\mathbf{K}_{cu}$ and $\mathbf{K}_{uc}$, and the coupling on its own. These matrices respectively account for the impact of the body stress state on the corrosion diffusion, and vice-versa.

Equation (4.8) shows that the matrices of $\mathbf{K}_{cu}$ and $\mathbf{K}_{uc}$ can be found respectively through expressions of nodal loads $\mathbf{g}_c$ and $\mathbf{g}_u$. In other words, we will start from the diffusion of corrosion equation to derive $\mathbf{g}_c$, and determine $\mathbf{K}_{cu}$ by identification. Same procedure for $\mathbf{K}_{uc}$ that will be determined starting from the mechanical behaviour equations.

Note finally, that since equation (4.1) coupled with (4.7) involves a partial time derivative, the global differential linear system may be expressed in all generality by

$$\begin{bmatrix} \mathbf{C}_{cc} & \mathbf{C}_{cu} \\ \mathbf{C}_{uc} & \mathbf{C}_{uu} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}}_c \\ \dot{\mathbf{q}}_u \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{cc} & \mathbf{K}_{cu} \\ \mathbf{K}_{uc} & \mathbf{K}_{uu} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_c \\ \mathbf{q}_u \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_c \\ \mathbf{g}_u \end{Bmatrix} \tag{4.9}$$

Assuming an integration scheme (as done in Chapter 1 and 2 where we opted for an Euler-implicit discretization scheme), this system degenerates after in linear system $\mathbf{Aq} = \mathbf{f}$ to solve, as stated by (4.8). If we did not used explicitly the damping matrix $\mathbf{C}$ in previous Chapter 1 and 2, it has well been considered implicitly (integrated to the stiffness matrix). The presentation shown in (4.9) will be preferred in this chapter, because it is more general, and enables therefore to use other integration schemes with unchanged expressions of stiffness matrix and nodal load vector (even if euler-implicit is only used in this report). It also highlights what is related to "speed" $\dot{\mathbf{q}}$ and "displacement" $\mathbf{q}$.

4.2.2 Elementary Stiffness matrix for corrosion effects

Some adjustments regarding the notation have been done in order to prevent any confusion within displacement and corrosion rate nodal unknowns (renamed $\mathbf{q}_c$ and $\mathbf{q}_u$ respectively). We will do the same for the shape function vectors/matrices $\mathbf{N}$ that will be written $\mathbf{N}_c$ and $\mathbf{N}_u$.

As done in the previous chapter, we will seek for an approximation function $\tilde{\mathbf{u}}$ for $\mathbf{u}$ and an approximation function $\tilde{c}$ for c. We remind here the expressions of $\tilde{\mathbf{u}}$, $\boldsymbol{\varepsilon}$ and $\boldsymbol{\sigma}$ known by the Chapter 3.

$$\mathbf{u}(x,y) = \begin{Bmatrix} u(x,y) \\ v(x,y) \end{Bmatrix} \approx \tilde{\mathbf{u}}(x,y) = \begin{Bmatrix} \tilde{u}(x,y) \\ \tilde{v}(x,y) \end{Bmatrix} = \mathbf{N}_u \mathbf{q}_u$$

$$\boldsymbol{\varepsilon} = \mathbf{B}\mathbf{q}_u \qquad\qquad \boldsymbol{\sigma} = \mathbf{D}\left(\mathbf{B}\mathbf{q}_u - \boldsymbol{\varepsilon_0}\right) + \boldsymbol{\sigma}_0$$

While for the corrosion scalar field $c(x, y)$, the following relationships have been established in Chapter 2.

$$c(x, y) \approx \tilde{c}(x, y) = \mathbf{N}_c \mathbf{q}_c$$

$$\frac{\partial \tilde{c}}{\partial x} = \mathbf{B}_x \mathbf{q}_c \qquad\qquad \frac{\partial \tilde{c}}{\partial y} = \mathbf{B}_y \mathbf{q}_c$$

Starting from the weak form (4.6), we get by substitution

$$\begin{aligned} e\ \delta\mathbf{q}_c^T \int_\Omega \mathbf{N}_c^T \mathbf{N}_c\ d\Omega\ \dot{\mathbf{q}_c} + eD\delta\mathbf{q}^T \int_\Omega \mathbf{B}_x^T \mathbf{B}_x \mathbf{q}_c + \mathbf{B}_y^T \mathbf{B}_y \mathbf{q}_c\ d\Omega \\ - eD \int_\Omega M\mathbf{B}_x^T \frac{\partial p}{\partial x}\mathbf{N}\mathbf{q}_c + M\mathbf{B}_y^T \frac{\partial p}{\partial y}\mathbf{N}\mathbf{q}_c\ d\Omega = 0 \end{aligned} \tag{4.10}$$

With $p = \dfrac{1}{2}\left(\sigma_{xx} + \sigma_{yy}\right)$. Developing the expressions of stresses:

$$p = \underbrace{\frac{1}{2} \cdot \begin{bmatrix} 1 & 1 & 0 \end{bmatrix}}_{=\mathbf{m}} \left[\mathbf{D}\left(\mathbf{B}\mathbf{q} - \boldsymbol{\varepsilon}_0\right) + \boldsymbol{\sigma}_0\right] \tag{4.11}$$

Hence, the spatial derivatives of pressure may be expressed as,

$$\frac{\partial p}{\partial x} = \mathbf{m}\left[\mathbf{D}\left(\frac{\partial \mathbf{B}}{\partial x}\mathbf{q}_u - \frac{\partial \boldsymbol{\varepsilon_0}}{\partial x}\right) + \frac{\partial \boldsymbol{\sigma}_0}{\partial x}\right] \tag{4.12}$$

$$\frac{\partial p}{\partial y} = \mathbf{m}\left[\mathbf{D}\left(\frac{\partial \mathbf{B}}{\partial y}\mathbf{q}_u - \frac{\partial \boldsymbol{\varepsilon_0}}{\partial y}\right) + \frac{\partial \boldsymbol{\sigma}_0}{\partial y}\right] \tag{4.13}$$

If we assume that the initial state does not involve any gradient in initial strain or stress, theses equations may be simplified as follow

$$\frac{\partial p}{\partial x} = \mathbf{m}\mathbf{D}\frac{\partial \mathbf{B}}{\partial x}\mathbf{q}_u = \mathbf{m}\mathbf{D}\mathbf{L}_x\mathbf{q}_u \tag{4.14}$$

$$\frac{\partial p}{\partial y} = \mathbf{m}\mathbf{D}\frac{\partial \mathbf{B}}{\partial y}\mathbf{q}_u = \mathbf{m}\mathbf{D}\mathbf{L}_y\mathbf{q}_u \tag{4.15}$$

where we have introduced the matrices $\mathbf{L}_x$ and $\mathbf{L}_y$ as the first derivatives of the strain matrix $\mathbf{B}$ with respect to x and y respectively. Substituting back the gradients of pressure in (4.10), it is obtained

$$\begin{aligned} e\ \delta\mathbf{q}_c^T \int_\Omega \mathbf{N}_c^T \mathbf{N}_c\ d\Omega\ \dot{\mathbf{q}_c} + eD\delta\mathbf{q}_c^T \int_\Omega \mathbf{B}_x^T \mathbf{B}_x \mathbf{q}_c + \mathbf{B}_y^T \mathbf{B}_y \mathbf{q}_c\ d\Omega \\ - eD\delta\mathbf{q}_c^T \int_\Omega M\mathbf{B}_x^T \mathbf{m}\mathbf{D}\mathbf{L}_x\mathbf{q}_u\mathbf{N}\mathbf{q}_c + M\mathbf{B}_y^T \mathbf{m}\mathbf{D}\mathbf{L}_y\mathbf{q}_u\mathbf{N}\mathbf{q}_c\ d\Omega = 0 \end{aligned} \tag{4.16}$$

This equation may be even more simplified, noting that the weighting coefficients $\delta\mathbf{q}_c^T$ are arbitrary,

$$e\int_\Omega \mathbf{N}_c^T\mathbf{N}_c\, d\Omega\, \dot{\mathbf{q}}_c + eD\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y\, d\Omega\mathbf{q}_c \\ - eDM\int_\Omega \mathbf{B}_x^T\mathbf{mDL}_x\boxed{\mathbf{q}_u}\mathbf{N}\boxed{\mathbf{q}_c} + \mathbf{B}_y^T\mathbf{mDL}_y\boxed{\mathbf{q}_u}\mathbf{N}\boxed{\mathbf{q}_c}\, d\Omega \tag{4.17}$$

This last equation raises an interesting issue: in the second line appears a product of both nodal unknowns $\mathbf{q}_u$ and $\mathbf{q}_c$. Providing that no additional simplification can be done, it is impossible to build a linear system such as the one stated at (4.8), which would have required a linear combination of $\mathbf{q}_c$ and $\mathbf{q}_u$ (instead of this, we get a product of $\mathbf{q}_u$ and $\mathbf{q}_c$). It results that the system formed by the second line of (4.17) can be solved only if either $\mathbf{q}_u$ or $\mathbf{q}_c$ are known. However, none of them are known at time t. Thus, the system defining the full coupling between corrosion and mechanical behavior is non linear. In order to get rid off one of the two unknowns, it is proposed to linearize the system, assuming that one of the unknowns $\mathbf{q}_u$ or $\mathbf{q}_c$ appearing in the product is identical to the previous iteration (then either[1] $\mathbf{q}_u^t = \mathbf{q}_u^{t+1}$ either $\mathbf{q}_c^t = \mathbf{q}_c^{t+1}$). At this stage, a first hypothesis could be made, but it is difficult to say which between the two mentioned options will be closer to the reality or even if both approaches are feasible. Hence the most reasonable choice is to examine them both, and decide after which one gives the most satisfactory results. Hence, two cases will be detailed in this report.

Option #1

Option #1 relies on the assumption that $\mathbf{q}_c^t = \mathbf{q}_c^{t+1}$ for concerned terms of (4.17). Accounting for this hypothesis and permuting the scalar quantities, we get

$$e\int_\Omega \mathbf{N}_c^T\mathbf{N}_c\, d\Omega\, \dot{\mathbf{q}}_c^t + eD\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y\, d\Omega\, \mathbf{q}_c^t \\ - eDM\int_\Omega \mathbf{B}_x^T\mathbf{N}\mathbf{q}_c^{t-1}\mathbf{mDL}_x + \mathbf{B}_y^T\mathbf{N}\mathbf{q}_c^{t-1}\mathbf{mDL}_y\, d\Omega\, \mathbf{q}_u^t \tag{4.18}$$

This equation may be rewritten as

$$\begin{bmatrix}\mathbf{C}_{e,cc} & \mathbf{0}\end{bmatrix}\begin{Bmatrix}\dot{\mathbf{q}}_c \\ \dot{\mathbf{q}}_u\end{Bmatrix} + \begin{bmatrix}\mathbf{K}_{e,cc} & \mathbf{K}_{e,cu}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_c \\ \mathbf{q}_u\end{Bmatrix} = \mathbf{0} \tag{4.19}$$

[1]Note that **only** the terms containing both $\mathbf{q}_u$ and $\mathbf{q}_c$ in a product are concerned with this hypothesis. (i.e. the framed quantities in (4.16)).

Where

$$\mathbf{C}_{e,cc} = \int_\Omega \mathbf{N}_c^T \mathbf{N}_c \, d\Omega \tag{4.20}$$

$$\mathbf{K}_{e,cc} = eD \int_\Omega \mathbf{B}_x^T \mathbf{B}_x + \mathbf{B}_y^T \mathbf{B}_y \, d\Omega \tag{4.21}$$

$$\mathbf{K}_{e,cu} = -eDM \int_\Omega \mathbf{B}_x^T \mathbf{N} \mathbf{q}_c^{t-1} \mathbf{m} \mathbf{D} \mathbf{L}_x + \mathbf{B}_y^T \mathbf{N} \mathbf{q}_c^{t-1} \mathbf{m} \mathbf{D} \mathbf{L}_y \, d\Omega \tag{4.22}$$

$$\mathbf{g}_{e,c} = \mathbf{0} \tag{4.23}$$

This system of equations is underspecified and needs to be supplemented by a second system. By identification with (4.9), it is observed that (4.19) constitutes the first equation of (4.9). The second equation of (4.9) should overcome this lack.

We shall see in the next section 4.2.3 that the constitutive matrix $\mathbf{D}$ is function of the corrosion state (and more precisely of a damage variable ϕ). If we assume $\mathbf{D}$ to be function of $\mathbf{q}^t$ instead of $\mathbf{q}^{t-1}$, a product of $\mathbf{q}_u^t$ and $\mathbf{q}_c^t$ will be formed twice in (4.18), in which case the formed system could not be linearized (thus, this is another source of nonlinearity). As a consequence, we will assume that the constitutive matrix $\mathbf{D}$ is evaluated at the previous time step $t-1$.

Option #2

The hypothesis carried out by the option #1 has led to a determination of $\mathbf{C}_{e,cc}$, $\mathbf{K}_{e,cc}$, $\mathbf{K}_{e,cu}$ and $\mathbf{g}_{e,c}$. We will repeat here exactly the same process, but assuming the complementary hypothesis, to obtain different expressions of the same matrices.

The second option assumes that $\mathbf{q}_u$ doesn't change much within Δt and then that $\mathbf{q}_u^t \approx \mathbf{q}_u^{t-1}$. Applied to (4.17), one have

$$\begin{aligned} e \int_\Omega \mathbf{N}_c^T \mathbf{N}_c \, d\Omega \dot{\mathbf{q}}_c + eD \int_\Omega \mathbf{B}_x^T \mathbf{B}_x + \mathbf{B}_y^T \mathbf{B}_y \, d\Omega \, \mathbf{q}_c^t \\ - eDM \int_\Omega \mathbf{B}_x^T \mathbf{m} \mathbf{D} \mathbf{L}_x \mathbf{q}_u^{t-1} \mathbf{N} + \mathbf{B}_y^T \mathbf{m} \mathbf{D} \mathbf{L}_y \mathbf{q}_u^{t-1} \mathbf{N} \, d\Omega \, {\mathbf{q}_c}^t \end{aligned} \tag{4.24}$$

That becomes, under "matrix" form

$$\begin{bmatrix} \mathbf{C}_{e,cc} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}}_c \\ \dot{\mathbf{q}}_u \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{e,cc} & \mathbf{K}_{e,cu} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_c \\ \mathbf{q}_u \end{Bmatrix} = \mathbf{0} \tag{4.25}$$

With

$$\begin{aligned}\mathbf{K}_{e,cc} &= eD\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y\, d\Omega \\ &\quad - eDM\int_\Omega \mathbf{B}_x^T\mathbf{mDL}_x\mathbf{q}_u^{t-1}\mathbf{N} + \mathbf{B}_y^T\mathbf{mDL}_y\mathbf{q}_u^{t-1}\mathbf{N}\, d\Omega\end{aligned} \tag{4.26}$$

$$\mathbf{K}_{e,cu} = \mathbf{0} \tag{4.27}$$

$$\mathbf{C}_{e,cc} = e\int_\Omega \mathbf{N}_c^T\mathbf{N}_c\, d\Omega \tag{4.28}$$

Here again, the constitutive matrix $\mathbf{D}$ can create a source of non-linearity. If $\mathbf{D}$ is evaluated at time t instead of $t-1$, the system (4.24) would be quadratic in $\mathbf{q}_c^t$. This assumption is identical to the one made in the option #1. Thus, to obtain the stiffness and damping matrices stated above, we had to assume that $\mathbf{q}_u^t = \mathbf{q}_u^{t-1}$ in the first instance, and $\mathbf{q}_c^t = \mathbf{q}_c^{t+1}$ in the second one. The assumption $\mathbf{q}_u^t = \mathbf{q}_u^{t-1}$ is thus **not self-sufficient to linearize the system and needs to be supplemented** by the second hypothesis $\mathbf{q}_c^t = \mathbf{q}_c^{t+1}$.

4.2.3 Elementary stiffness matrix for mechanics

Restarting from the weak form stated at section 4.1.2, the weighted residual is given by

$$\mathtt{WR} = \int_V \delta\mathbf{q}_u^T\mathbf{B}^T\mathbf{DB}\mathbf{q}_u\, dV - \int_V \delta\mathbf{q}_u^T\mathbf{N}^T\mathbf{b}\, dV = 0 \tag{4.29}$$

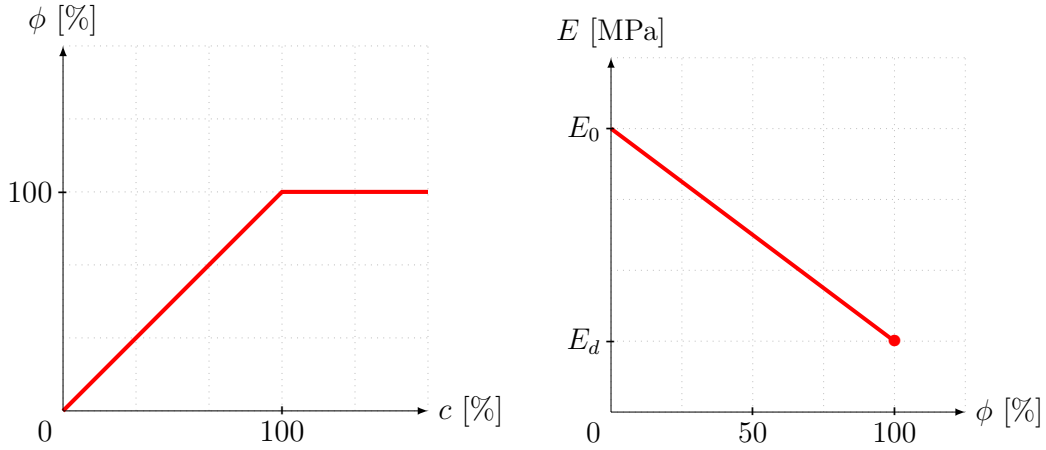
$$\Leftrightarrow \mathtt{WR} = \delta\mathbf{q}_u^T e\left(\int_\Omega \mathbf{B}^T\mathbf{DB}\, d\Omega\, \mathbf{q}_u - \int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega\right) = 0 \tag{4.30}$$

Because $\delta\mathbf{q}_u^T$ are arbitrary, the previous equation degenerates into the linear system:

$$e\int_\Omega \mathbf{B}^T\mathbf{DB}\, d\Omega\, \mathbf{q}_u = e\int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega \tag{4.31}$$

In this coupling model, we will assume that corrosion rate c influences the material properties. Therefore, the Hooke's matrix $\mathbf{D}$ is no longer constant. We will assume in the first instance that only the Young modulus E will vary as a function of c. To model this effect, we suggest to introduce a damage variable ϕ, strongly related to c. The higher the c (or ϕ), the lower the Young modulus: any considered damage law should then be a strictly decreasing function in c. The easiest law to incorporate in our model is a linear law such as suggested in Figure 4.1, expressed mathematically as

$$E(\phi) = E_0 \cdot \left[\left(1 - \frac{E_d}{E_0}\right)\cdot(1-\phi) + \frac{E_d}{E_0}\right] = E_0\cdot[1 + \phi\cdot(\alpha - 1)] = E_0\cdot f^\star(\phi) \tag{4.32}$$



(a) Definition of the damage variable ϕ with respect to c

(b) Evolution of Young modulus E as a function of the damage ratio c for a linear damage law

Figure 4.1: Description of damage variable ϕ and the chosen damage law

With $\alpha = E_d/E_0$ and $f^\star(\phi) = 1 + \phi \cdot (\alpha - 1)$. The Hooke's matrix is now given by

$$\mathbf{D} = f^\star(\phi) \cdot \underbrace{\frac{E_0}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & (1-\nu)/2 \end{bmatrix}}_{= \mathbf{D}_0} = f^\star(\phi) \cdot \mathbf{D}_0 \tag{4.33}$$

We now understand why it is required to make use of an intermediate variable ϕ instead of c directly: the relative oxygen concentration $c = [0_2]/[0_2]_{cr}$ can take values higher than 1, depending on the value chosen for $[0_2]_{cr}$ and lead to negative Young modulus (or lower than residual Young modulus). The damage variable ϕ is however equal to c in the bounded interval $[0,\ 1]$ and admits 0 and 1 respectively as minimal and maximal value.

The function $f^\star(\phi)$ can be determined with respect to the nodal unknowns $\mathbf{q}_c$

$$f^\star(\phi) = 1 + \phi \cdot (\alpha - 1) \Rightarrow f^\star(\xi, \eta) = 1 + (\alpha - 1) \cdot \mathbf{N}_c \mathbf{q}_\phi \tag{4.34}$$

with

$$\mathbf{q}_\phi = \min\left(\mathbf{q}_c,\ 1\right) \tag{4.35}$$

And the constitutive matrix takes the following expression

$$\mathbf{D} = \left[1 + (\alpha - 1) \cdot \mathbf{N}_c \min\left(\mathbf{q}_c,\ \mathbf{1}\right)\right] \mathbf{D}_0 \tag{4.36}$$

At this stage, two problems must be pointed out.

- The first problem has been briefly discussed previously. Equation (4.36) shows that the matrix $\mathbf{D}$ is function of $\mathbf{q}_c$. If the constitutive matrix at time t is evaluated with the variable $\mathbf{q}_c$ at the same time, i.e. if we assume that

$$\mathbf{D}^t = \left[1 + (\alpha - 1) \cdot \mathbf{N}_c \min\left(\mathbf{q}_c^t,\ \mathbf{1}\right)\right] \mathbf{D}_0$$

equations (4.18) and (4.24), become both non-linear. As our aim in this work is to develop a linear model, it is suggested that $\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$ and therefore

$$\mathbf{D}^t = \left[1 + (\alpha - 1) \cdot \mathbf{N}_c \min\left(\mathbf{q}_c^{t-1},\ \mathbf{1}\right)\right] \mathbf{D}_0 \tag{4.37}$$

which eliminates the non-linearity. This explains why we assumed in previous section that $\mathbf{D}$ should be evaluated with Young's modulus from previous time.

- Equation (4.36) involves the min operator applied to the unknown variable $\mathbf{q}_c$ (and unit vector). This operator is non-linear as well. However, this problem is fixed by the assumption made previously: if we assume $\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$, the minimum operator is no more applied to the nodal unknown $\mathbf{q}_c^t$ itself but on its previous value $\mathbf{q}_c^{t-1}$. As a consequence, the damage variable at nodes at time t will be expressed by

$$\mathbf{q}_\phi^t = \min\left(\mathbf{N}_c \mathbf{q}_c^{t-1}, \mathbf{1}\right) \tag{4.38}$$

Now that the expression of constitutive matrix $\mathbf{D}$ is known, it can be substituted into (4.31). We get

$$e\left[1 + (\alpha - 1)\right] \int_\Omega \mathbf{B}^T \mathbf{N}_c \mathbf{q}_\phi^t \mathbf{D}_0 \mathbf{B}\ d\Omega\ \mathbf{q}_u = e \int_\Omega \mathbf{N}^T \mathbf{b}\ d\Omega \tag{4.39}$$

This last equation can be written as

$$\begin{bmatrix} \mathbf{K}_{e,uc} & \mathbf{K}_{e,cc} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_{e,c} \\ \mathbf{q}_{e,u} \end{Bmatrix} = \mathbf{g}_{e,u} \tag{4.40}$$

And by identification with (4.9), it is found that

$$\mathbf{C}_{e,uc} = \mathbf{0} \tag{4.41}$$

$$\mathbf{C}_{e,uu} = \mathbf{0} \tag{4.42}$$

$$\mathbf{K}_{e,uc} = \mathbf{0} \tag{4.43}$$

$$\mathbf{K}_{e,uu} = e\left[1 - (1 - \alpha)\right] \int_\Omega \mathbf{B}^T \mathbf{N}_c \mathbf{q}_\phi^t \mathbf{D}_0 \mathbf{B}\ d\Omega \tag{4.44}$$

$$\mathbf{g}_{e,u} = e \int_\Omega \mathbf{N}^T \mathbf{b}\ d\Omega \tag{4.45}$$

Thus, options #1 and #2 admit common expressions for the stiffness/damping matrices and nodal load vector given above. These expressions (4.41)-(4.45) have been obtained using one single assumption: $\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$.

4.2.4 Elementary coupling equations system

The mathematical developments hold in the two previous sections may be assembled and summarized by building the linear differential system:

$$\begin{bmatrix} \mathbf{C}_{cc} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \dot{\mathbf{q}}_{e,c} \\ \dot{\mathbf{q}}_{e,u} \end{Bmatrix} + \begin{bmatrix} \mathbf{K}_{e,cc} & \mathbf{K}_{e,cu} \\ \mathbf{K}_{e,uc} & \mathbf{K}_{e,uu} \end{bmatrix} \begin{Bmatrix} \mathbf{q}_{e,c} \\ \mathbf{q}_{e,u} \end{Bmatrix} = \begin{Bmatrix} \mathbf{g}_{e,c} \\ \mathbf{g}_{e,u} \end{Bmatrix} \tag{4.46}$$

The expressions of stiffness/damping matrices and nodal load vectors are summarized in Table 4.1 for both examined options. Note that the hypothesis of constant diffusivity may easily be relaxed putting D inside the integral.

We now have two differents linear differential systems of equations to model the coupling between corrosion and mechanics. Let us now discuss the qualitative differences between the two options.

In the first option, one single hypothesis has been made to linearize the system ($\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$), while linear terms remained unaffected by this assumption. The assumption was self-sufficient and did not have to be supplemented by an additional condition, by contrast with the option #2, for which the assumption $\mathbf{q}_u^t \approx \mathbf{q}_u^{t-1}$ was not sufficient to lead to a linear system. A linearized system could be reached for the option #2 at the expense of a second hypothesis: $\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$, which was assumed in the first approach. Henceforth, the first scheme is less constrained than the second, and it appears to be more consistent. For these reasons, we will prefer the coupling model derived in option #1 rather than in option #2.

	Option #1	**Option #2**
Assumptions	$\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$	$\mathbf{q}_u^t \approx \mathbf{q}_u^{t-1}$ and $\mathbf{q}_c^t \approx \mathbf{q}_c^{t-1}$
$\mathbf{C}_{cc}$	$e\int_\Omega \mathbf{N}_c^T\mathbf{N}_c\, d\Omega$	$e\int_\Omega \mathbf{N}_c^T\mathbf{N}_c\, d\Omega$
$\mathbf{K}_{e,cc}^t$	$eD\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y\, d\Omega$	$eD\int_\Omega \mathbf{B}_x^T\mathbf{B}_x + \mathbf{B}_y^T\mathbf{B}_y - M\mathbf{B}_x^T\mathbf{m}\mathbf{D}^t\mathbf{L}_x\mathbf{q}_u^{t-1}\mathbf{N} - M\mathbf{B}_y^T\mathbf{m}\mathbf{D}^t\mathbf{L}_y\mathbf{q}_u^{t-1}\mathbf{N}\, d\Omega$
$\mathbf{K}_{e,cu}^t$	$-eDM\int_\Omega \mathbf{B}_x^T\mathbf{N}\mathbf{q}_c^{t-1}\mathbf{m}\mathbf{D}\mathbf{L}_x + \mathbf{B}_y^T\mathbf{N}\mathbf{q}_c^{t-1}\mathbf{m}\mathbf{D}\mathbf{L}_y\, d\Omega$	$\mathbf{0}$
$\mathbf{K}_{e,uc}^t$	$\mathbf{0}$	$\mathbf{0}$
$\mathbf{K}_{e,uu}$	$e\left[1+(\alpha-1)\right]\int_\Omega \mathbf{B}^T\mathbf{N}_c\mathbf{q}_c^{t-1}\mathbf{D}_0\mathbf{B}\, d\Omega$	$e\left[1+(\alpha-1)\right]\int_\Omega \mathbf{B}^T\mathbf{N}_c\mathbf{q}_c^{t-1}\mathbf{D}_0\mathbf{B}\, d\Omega$
$\mathbf{g}_{e,c}$	$\mathbf{0}$	$\mathbf{0}$
$\mathbf{g}_{e,u}$	$e\int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega$	$e\int_\Omega \mathbf{N}^T\mathbf{b}\, d\Omega$

$$\begin{bmatrix}\mathbf{C}_{cc} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\begin{Bmatrix}\dot{\mathbf{q}}_{e,c}\\ \dot{\mathbf{q}}_{e,u}\end{Bmatrix} + \begin{bmatrix}\mathbf{K}_{e,cc} & \mathbf{K}_{e,cu}\\ \mathbf{K}_{e,uc} & \mathbf{K}_{e,uu}\end{bmatrix}\begin{Bmatrix}\mathbf{q}_{e,c}\\ \mathbf{q}_{e,u}\end{Bmatrix} = \begin{Bmatrix}\mathbf{g}_{e,c}\\ \mathbf{g}_{e,u}\end{Bmatrix}$$

Table 4.1: Summary of the expressions of elementary stiffness matrices, damping matrices and consistent nodal load vectors for both examined options

4.3 Choice of shape functions

4.3.1 Position of the problem

Now that the expressions of stiffness and damping matrices are known, we must address the question of the shape functions. In the two first chapters 1 and 2, we only considered linear shape functions as they meet the minimal requirements so that the convergence is assured. We have proven that the same development can be applied to the displacement based model developed in Chapter 3. One perfectly legitimate question may be asked at that point as if it worked well for the 3 previous chapters, why should we reconsider the question of the shape functions. The answer can be found in the additional non-fickian flux which has been added in the diffusion equation (4.1), and consequently in the weak form (4.6)

$$e\int_\Omega \frac{\partial c}{\partial t}\delta c\, d\Omega + eD\int_\Omega \frac{\partial c}{\partial x}\cdot\frac{\partial \delta c}{\partial x} + \frac{\partial c}{\partial y}\cdot\frac{\partial \delta c}{\partial y} - Mc\frac{\partial p}{\partial x}\cdot\frac{\partial \delta c}{\partial x} - Mc\frac{\partial p}{\partial y}\cdot\frac{\partial \delta c}{\partial y}\, d\Omega = 0$$

This latter incorporates a gradient of pressure, which is itself a first derivative of the displacement field $\mathbf{u}$. Therefore, a second order derivative is integrated to the stiffness matrix expressions (characterized by $\mathbf{L}_x$ and $\mathbf{L}_y$ in Table 4.1).

Reminding from [25, 37, 38] that for elasticity problems, the convergence is ensured if both the continuity (or compatibility) and the completeness requirements are achieved. By continuity, it is understood that the continuity of the unknown is enforced at the interface between two elements, while the slope continuity is not required. In other words, a $\mathbb{C}_0$ continuity must be enforced. The second requirement, the completeness, imposes that the shape functions are such that a constant strain may be modeled all inside an element. From these considerations, we concluded that linear shape functions in ξ and η, (or of higher degree polynomial shape functions) were suitable for both corrosion and mechanical models.

Let us assume that linear shape functions identical to the ones derived in section 1.2.5 a appropriate for this coupling model. The action of the non-fickian flux are translated in the linear differential system (4.46) by the matrix $\mathbf{K}_{e,cu}$ whose expression is given by 4.22:

$$\mathbf{K}_{e,cu} = -eDM\int_\Omega \mathbf{B}_x^T\mathbf{N}\mathbf{q}_c^{t-1}\mathbf{m}\mathbf{D}\mathbf{L}_x + \mathbf{B}_y^T\mathbf{N}\mathbf{q}_c^{t-1}\mathbf{m}\mathbf{D}\mathbf{L}_y\, d\Omega$$

It can be observed that one first derivative with respect to x or y of the strain matrix appears in both terms of this equation. Indeed, if we focus on the first

term for example, we get by definition of the matrix **B** 3.30,

$$\mathbf{L}_x = \frac{\partial}{\partial x}\mathbf{B}(\xi,\eta)$$

$$= \begin{bmatrix} \frac{\partial^2 N_1}{\partial x^2} & 0 & \frac{\partial^2 N_2}{\partial x^2} & 0 & \frac{\partial^2 N_3}{\partial x^2} & 0 & \frac{\partial^2 N_4}{\partial x^2} & 0 \\ 0 & \frac{\partial^2 N_1}{\partial x \partial y} & 0 & \frac{\partial^2 N_2}{\partial x \partial y} & 0 & \frac{\partial^2 N_3}{\partial x \partial y} & 0 & \frac{\partial^2 N_4}{\partial x \partial y} \\ \frac{\partial^2 N_1}{\partial x \partial y} & \frac{\partial^2 N_1}{\partial x^2} & \frac{\partial^2 N_2}{\partial x \partial y} & \frac{\partial^2 N_2}{\partial x^2} & \frac{\partial^2 N_3}{\partial x \partial y} & \frac{\partial^2 N_3}{\partial x^2} & \frac{\partial^2 N_4}{\partial x \partial y} & \frac{\partial^2 N_4}{\partial x^2} \end{bmatrix}$$

Its components may be determined as follow,

$$= \frac{\partial}{\partial x}\left(\frac{\partial N_i}{\partial x}\right)$$

$$= \frac{\partial}{\partial x}\left(\frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial x}\right)$$

$$= \left(\frac{\partial}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial}{\partial \eta}\frac{\partial \eta}{\partial x}\right)\left(\frac{\partial N_i}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial N_i}{\partial \eta}\frac{\partial \eta}{\partial x}\right)$$

$$= \left(\frac{\partial \xi}{\partial x}\right)^2 \frac{\partial^2 N_i}{\partial \xi^2} + \left(\frac{\partial \eta}{\partial x}\right)^2 \frac{\partial^2 N_i}{\partial \eta^2} + \left(\frac{\partial \eta}{\partial x}\frac{\partial \xi}{\partial x}\right)\frac{\partial^2 N_i}{\partial \eta \partial \xi} + \left(\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial x}\right)\frac{\partial^2 N_i}{\partial \xi \partial \eta}$$

In virtue of the Schwarz's theorem,

$$\frac{\partial^2 N_i}{\partial x^2} = \left(\frac{\partial \xi}{\partial x}\right)^2 \frac{\partial^2 N_i}{\partial \xi^2} + \left(\frac{\partial \eta}{\partial x}\right)^2 \frac{\partial^2 N_i}{\partial \eta^2} + 2 \cdot \frac{\partial \eta}{\partial x}\frac{\partial \xi}{\partial x} \cdot \frac{\partial^2 N_i}{\partial \xi \partial \eta} \tag{4.47}$$

It is proven on the same manner that,

$$\frac{\partial^2 N_i}{\partial y^2} = \left(\frac{\partial \xi}{\partial y}\right)^2 \frac{\partial^2 N_i}{\partial \xi^2} + \left(\frac{\partial \eta}{\partial y}\right)^2 \frac{\partial^2 N_i}{\partial \eta^2} + 2 \cdot \frac{\partial \eta}{\partial y}\frac{\partial \xi}{\partial y} \cdot \frac{\partial^2 N_i}{\partial \xi \partial \eta} \tag{4.48}$$

$$\frac{\partial^2 N_i}{\partial x \partial y} = \frac{\partial \xi}{\partial x}\frac{\partial \xi}{\partial y} \cdot \frac{\partial^2 N_i}{\partial \xi^2} + \left(\frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} + \frac{\partial \xi}{\partial y}\frac{\partial \eta}{\partial x}\right) \cdot \frac{\partial^2 N_i}{\partial \xi \eta} + \frac{\partial \eta}{\partial x}\frac{\partial \eta}{\partial y} \cdot \frac{\partial^2 N_i}{\partial \eta^2} \tag{4.49}$$

It might be useful to remind that the factors $\frac{\partial(.)}{\partial x}$ and $\frac{\partial(.)}{\partial y}$ in the above formula are determined by identification with the inverse jacobian matrix 2.36.

$$\mathbf{J}^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix}$$

If linear shape functions are used, only the second "cross" derivative is non-zero, admitting a constant value everywhere inside the element.

$$\frac{\partial^2 N_i}{\partial \xi^2} = 0 \qquad \text{and} \qquad \frac{\partial^2 N_i}{\partial \eta^2} = 0 \qquad \text{and} \qquad \frac{\partial^2 N_i}{\partial \xi \partial \eta} = \text{Const.}$$

Hence, linear shape function are unable to represent any variation of gradient of pressure inside an element. If we now assume that the mesh is regular, then it comes that $\frac{\partial \xi}{\partial y} = \frac{\partial \eta}{\partial x} = 0$, which implies by (4.47) and (4.48) that $\frac{\partial^2 N_i}{\partial x^2} = \frac{\partial^2 N_i}{\partial y^2} = 0$. (4.49) is also reduced to

$$\frac{\partial^2 N_i}{\partial x \partial y} = \frac{\partial \xi}{\partial x}\frac{\partial \eta}{\partial y} \cdot \frac{\partial^2 N_i}{\partial \xi \eta} = J_{11}^{-1} J_{22}^{-1} \cdot \frac{\partial^2 N_i}{\partial \xi \eta} = \frac{1}{J}\frac{\partial^2 N_i}{\partial \xi \eta} = \frac{1}{\Omega}\frac{\partial^2 N_i}{\partial \xi \eta}$$

Where Ω is the area of the finite element. And thus, for linear shape functions the matrix $\mathbf{L}_x$ is

$$\mathbf{L}_x = \frac{1}{\Omega}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{4.50}$$

If the same process is applied to $\mathbf{L}_y$,

$$\mathbf{L}_y = \frac{1}{\Omega}\begin{bmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{4.51}$$

As it can be seen in previous equations, the matrices $\mathbf{L}_x$ and $\mathbf{L}_y$ are reduced to very simple forms and their integration in (4.22) will potentially result in a consequent loss of information. It seems indeed that linear shape functions may not be sufficiently rich to represent properly the evolution of the pressure gradient. The most relevant question being to ask is to quantify this lost of information, and to determine to which extend it can decrease the quality of the results. The answer is not an easy one, and the most appropriate approach to tackle the problem is to consider both linear and higher order shape functions. Opting for higher order shape functions will indeed fill the matrix $\mathbf{L}_x$ and $\mathbf{L}_y$ on a more complete manner, as their derivative are at least linear. Therefore, two distinct models have been developed. The first with linear shape functions (identical to the one exposed in previous chapters) and the second one with parabolic shape functions that will theoretically enrich the gradient of pressure polynomial expansion.

For compatibility issues, it is proposed to adopt the same element type indifferently to the problem considered (corrosion or mechanics of solids).

As the linear shape functions have already been detailed, we will not remind here the developments held in previous chapter. Only the second order shape functions will be covered by the next section.

4.3.2 Second order shape functions

Element types chosen for this second model are Lagrange nine-noded elements. In terms of behavior these elements are more efficient than Surendipity elements (8-nodes quadrangles) because of their ability to represent exactly any quadratic function. Indeed, Surendipity elements have an incomplete polynomial expansion [25] such that the term $\xi^2\eta^2$ is not represented:

$$u(\xi, \eta) = \beta_1 + \beta_2\xi + \beta_3\eta + \beta_4\xi^2 + \beta_5\eta^2 + \beta_6\eta\xi + \beta_7\xi\eta^2 + \beta_8\xi^2\eta$$

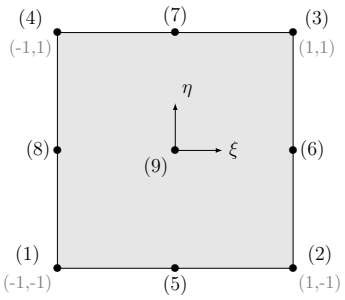


Figure 4.2: Lagrange 9-nodes element (`quad9`)

It follows that lagrangian, 9-node elements are more adapted to represent highly distorted elements and hence, model smoother solutions [25]. However, this come at the expense of computation speed, since this 9^{th} node adds 3 degrees of freedom by elements.

The lagrangian 9-noded element is shown in flat adimensional coordinate system (ξ, η) in Figure 4.2. The expressions of shape function are determined from quadratic Lagrange polynomials extended to two-dimensions. The obtained polynomials are

$$N_1(\xi,\eta) = \frac{1}{4}\cdot\xi\eta\cdot(\xi-1)(\eta-1) \tag{4.52a}$$

$$N_2(\xi,\eta) = \frac{1}{4}\cdot\xi\eta\cdot(\xi+1)(\eta-1) \tag{4.52b}$$

$$N_3(\xi,\eta) = \frac{1}{4}\cdot\xi\eta\cdot(\xi+1)(\eta+1) \tag{4.52c}$$

$$N_4(\xi,\eta) = \frac{1}{4}\cdot\xi\eta\cdot(\xi-1)(\eta+1) \tag{4.52d}$$

$$N_5(\xi,\eta) = \frac{1}{2}\cdot\eta\cdot(1-\xi^2)(\eta-1) \tag{4.52e}$$

$$N_6(\xi,\eta) = \frac{1}{2}\cdot\xi\cdot(1-\eta^2)(\xi+1) \tag{4.52f}$$

$$N_7(\xi,\eta) = \frac{1}{2}\cdot\eta\cdot(1-\xi^2)(\eta+1) \tag{4.52g}$$

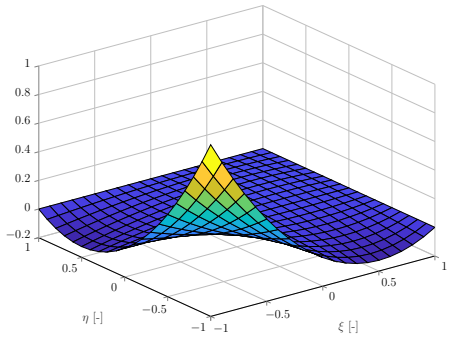
$$N_8(\xi,\eta) = \frac{1}{2}\cdot\xi\cdot(1-\eta^2)(\xi-1) \tag{4.52h}$$

$$N_9(\xi,\eta) = (1-\xi^2)(1-\eta^2) \tag{4.52i}$$

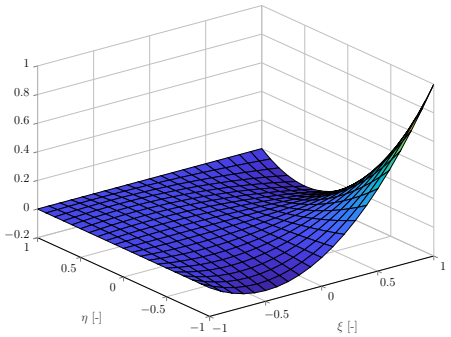
These shape functions are represented in Figure 4.3.

Matrices $\mathbf{B}$, $\mathbf{B}_x$, $\mathbf{B}_y$, $\mathbf{L}_x$ and $\mathbf{L}_y$ present in stiffness/damping matrices are derived on the same manner than shown previously. Only differs the matrix sizes. The size of stiffness and damping matrices are consequently also modified.

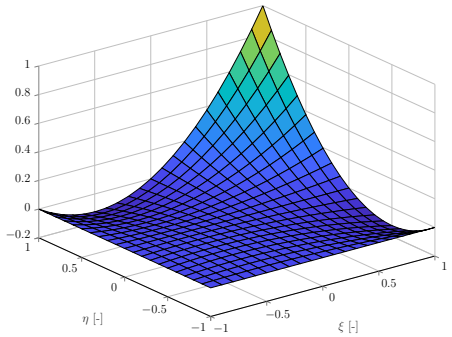
Finally, it is of interest to note that the upgrade of shape functions to second order polynomial requires a higher number of gauss points for the Gauss-Legendre quadrature. It is known by [25,35] that 3×3 Gauss points must be used in order to avoid formation of hourglass modes (or zero energy singular modes), which



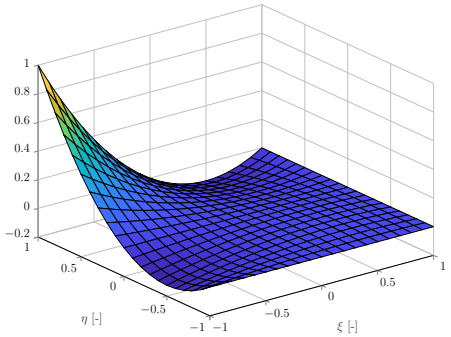
(a) Shape function $N_1(\xi, \eta)$



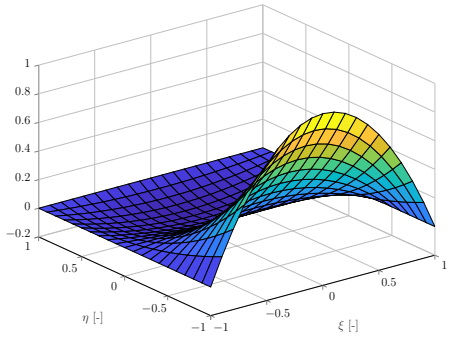
(b) Shape function $N_2(\xi, \eta)$



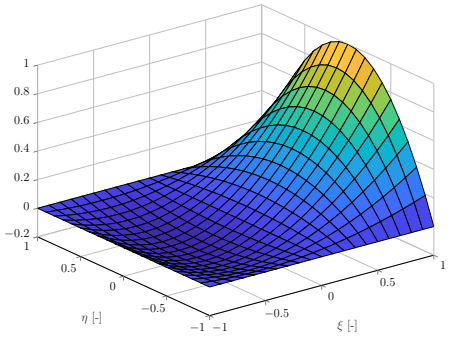
(c) Shape function $N_3(\xi, \eta)$



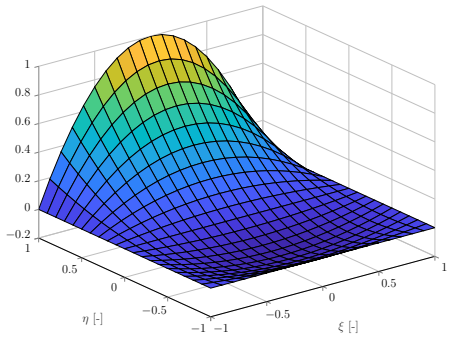
(d) Shape function $N_4(\xi, \eta)$



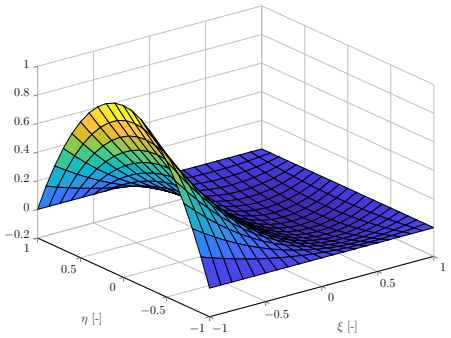
(e) Shape function $N_5(\xi, \eta)$



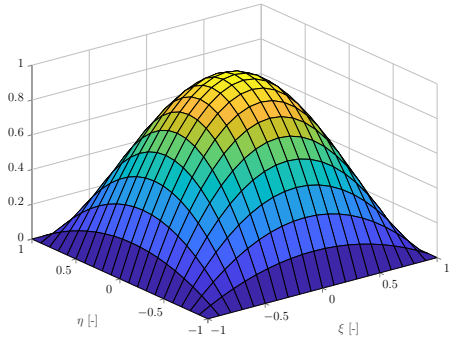
(f) Shape function $N_6(\xi, \eta)$



(g) Shape function $N_7(\xi, \eta)$



(h) Shape function $N_8(\xi, \eta)$



(i) Shape function $N_9(\xi, \eta)$

Figure 4.3: Shape functions for 9-node lagrangian element

further increases the computational cost with respect to linear shape functions, at equal number of elements.

4.4 System Resolution

Global stiffness/damping matrices and nodal load vectors are obtained performing an assembly operation as shown in (2.25) and (2.26). The global nodal unknowns $\mathbf{q}_c$ and $\mathbf{q}_u$ are obtained by solving the linear differential system (4.46), which is of the form

$$\mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{g} \qquad \text{with} \qquad \mathbf{q} = \begin{Bmatrix} \mathbf{q}_c \\ \mathbf{q}_u \end{Bmatrix} \tag{4.53}$$

Assuming an implicit first order discretization scheme for the partial derivative, we get

$$\mathbf{C}\left(\frac{\mathbf{q}^{t+1} - \mathbf{q}^t}{\Delta t}\right) + \mathbf{K}^{t+1}\mathbf{q}^{t+1} = \mathbf{g}^{t+1} \tag{4.54}$$

Which is a linear system for $\mathbf{q}^{t+1}$

$$(\mathbf{C} + \Delta t\mathbf{K}^{t+1})\mathbf{q}^{t+1} = \mathbf{C}\mathbf{q}^t + \Delta t\mathbf{g}^{t+1} \tag{4.55}$$

This integration schema may be qualified as implicit, as the right hand side of (4.54) is evaluated at $t+1$. But since the stiffness matrix $\mathbf{K}$ is function of $\mathbf{D}^{t+1}$, which itself evaluates the material properties from nodal unknown $\mathbf{q}_c$ at previous time step (see equation (4.37)), this is rather an approached implicit integration schema. A fully implicit integration scheme would have required an evaluation of the stiffness matrix at time $t+1$, relaxing the hypothesis $\mathbf{q}^{t+1} = \mathbf{q}^t$, and thus a non-linear system would have to be solved. A such resolution is performed with an iterative process such as Newton-Raphson method, evaluating the exact tangent stiffness matrix at each iteration.

Because our system has been linearized, its solution is directly obtained by "inverting" (4.55). But it must be beard in mind that the solution $\mathbf{q}^{t+1}$ is the solution for the linearized system. Henceforth, it shall be as away from the actual solution of the non-linear problem as the value chosen for Δt is indelicate (because the stiffness matrix will deviate from the tangent stiffness matrix as Δt increases). The error committed on the solution is hence highly dependent on Δt, and is a decreasing function of the chosen time step. If one wants to ensure the convergence of the numerical scheme, one same test case can be analyzed for decreasing values of Δt. Once the results are close to each others, we have reached trustful values of Δt (because convergence is observed).

To conclude, (4.55) shall be qualified as an approached implicit discretization scheme, and one can say that convergence can be reached if Δt is sufficiently small. It can be said that the scheme is stable unconditionally even if the existence of a potential stability criterion should be investigated. The solution found by (4.55) will match faithfully the solution of the non-linear, only provided that Δt tends to 0. Otherwise, deviations must be expected.

4.5 Numerical developments

An flow-chart summarizing the main steps of the numerical code implementation is shown in Figure 4.4.

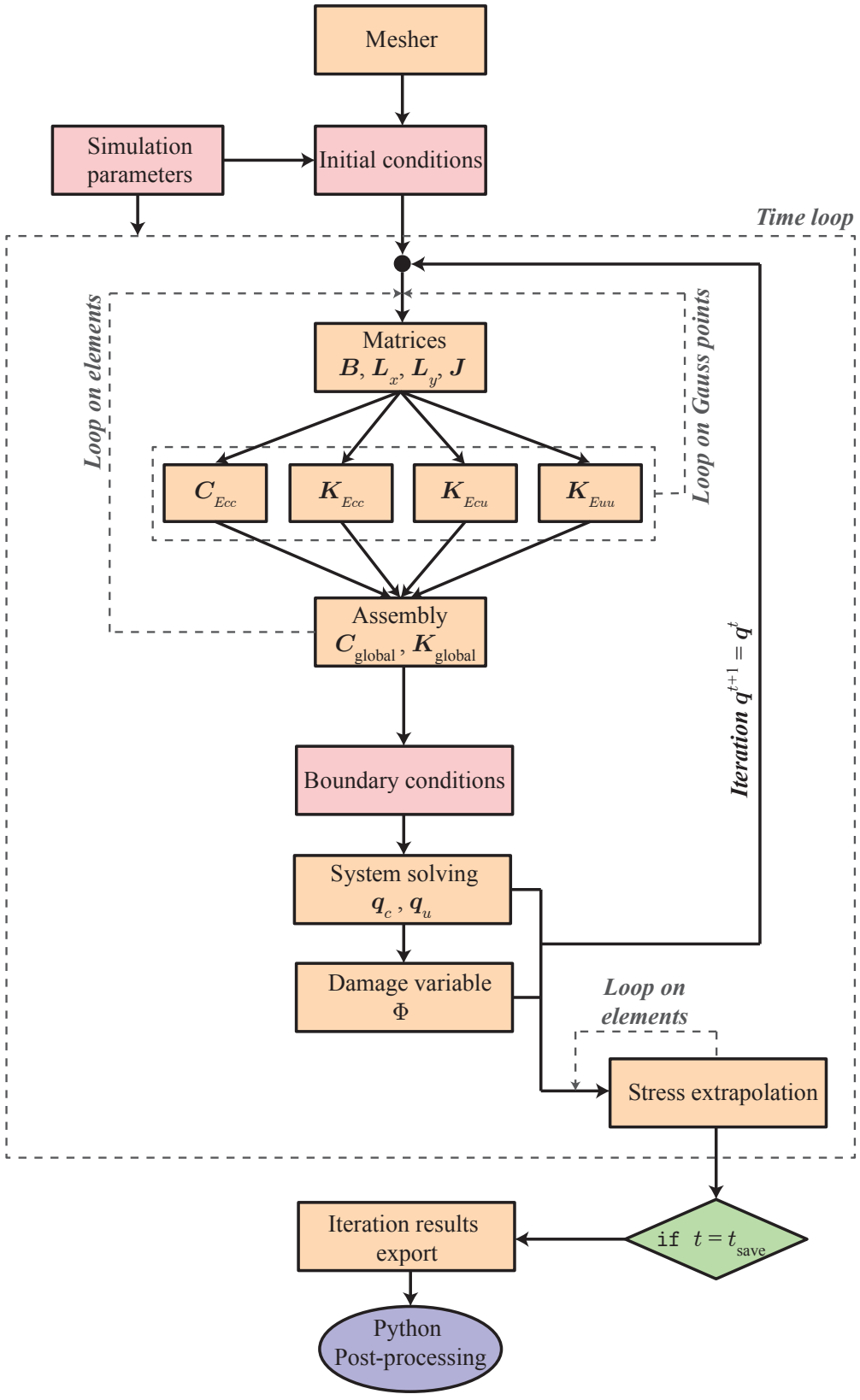


Figure 4.4: Flow-chart of the numerical code

4.6 Results

The analytical solution of the system defined by the two considered equations (3.1) and (2.3) is way too much complicated to be derived. Hence, no comparison with a reference solution will by carried out by this section. Instead, the results will be commented and see if they can be explained by the fundamental equations. Also, two models have been developed based on linear and quadratic shape functions. If the equations are properly solved, the 9-noded elements should provide more accurate results, at equal number of elements.

4.6.1 Test Case 1

For the first test case, it is proposed to test the influence of the corrosion on the mechanical problem. For this purpose, it will be assumed that the pressure factor $M = 0$, in which case the stiffness matrix $\mathbf{K}$ becomes diagonal and the model is reduced to a staggered coupling model. The situation consists in a rectangular plate in extension and submitted to a corrosion rate $c = \frac{1}{65} = 153.8\%$ on its left upper corner. The numerical results are shown in Figure 4.6 and Figure 4.8. It can

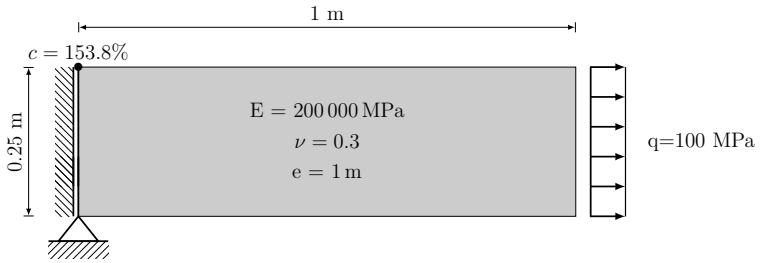
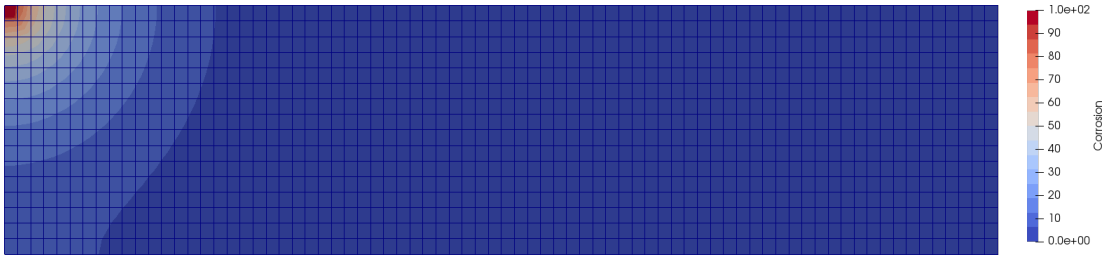


Figure 4.5: Description of test case 1.

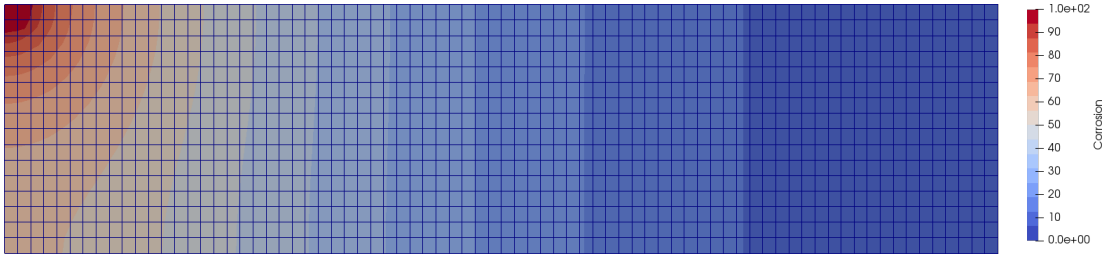
be seen in Figure 4.6 that the corrosion increases significantly the displacements at the right upper corner. The reason for this is that the corrosion induces a material softening (remind that a linear damage law was assumed). Initially, the isolines of displacements $u(x)$ are rather vertical and are progressively inclined $u(x)$ as the corrosion diffuses, proof of the gradient of stiffness acting all along the height of the left plate border.

In Figure 4.8, Von-Mises stresses have been plotted for 3 different times. The stresses reaches at its lowest were the corrosion rate is the highest. A redistribution analogous to a plastic redistribution occurs then all around that point: as the effort may not transit where the material is softened, it transits below and the stresses are then higher at the center of the left plate border. As the corrosion diffuses with the time increasing, the soft material zone is progressively enlarged and the peak of stresses is pushed down. The zone defined by this stress peak grows also gradually as the diffusion takes place, and maximal values of stress σ_{VM} increases in parallel. This redistribution modifies also the global behavior of the plate, which was initially under pure tension.

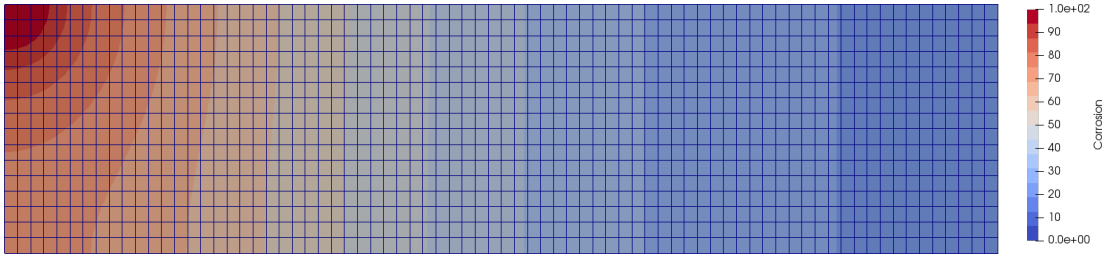
This non-uniform distribution of stresses σ_x along the border induces transverse displacements in the neighborhood of the left extremity. It is also to interest to note that since the tension stresses are lower at the upper right corner, they are



(a) $t = 0.15\,\mathrm{s}$

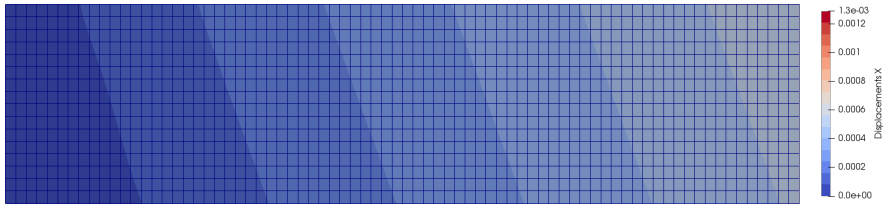


(b) $t = 1.8\,\mathrm{s}$

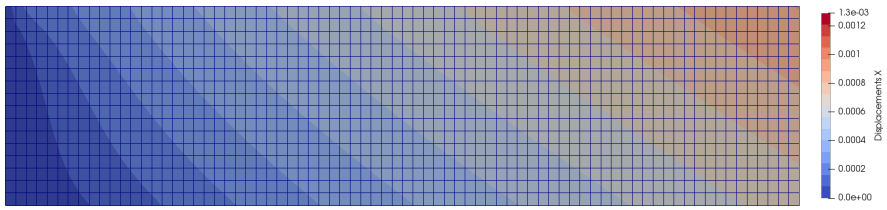


(c) $t = 3\,\mathrm{s}$

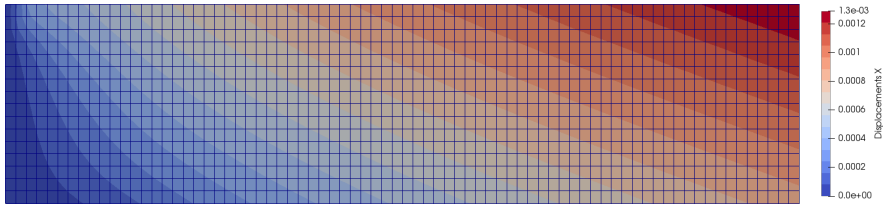
Figure 4.6: Damage variable ϕ [%] obtained for Test Case 1



(a) $t = 0.15\,\mathrm{s}$

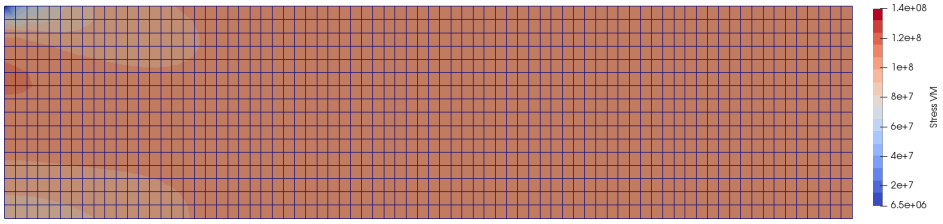


(b) $t = 1.8\,\mathrm{s}$

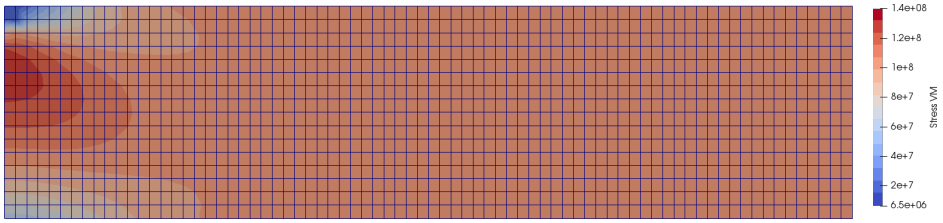


(c) $t = 3\,\mathrm{s}$

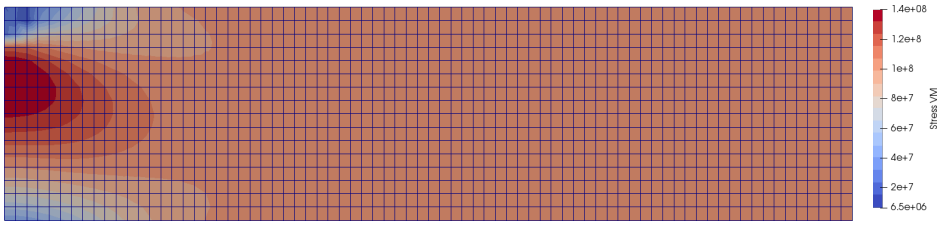
Figure 4.7: Displacement u[m] obtained for Test Case 1



(a) σ_{VM} [Pa] at $t = 0.15\,\mathrm{s}$



(b) σ_{VM} [Pa] at $t = 1.8\,\mathrm{s}$



(c) σ_{VM} [Pa] at $t = 3\,\mathrm{s}$

Figure 4.8: Von Mises stresses obtained for Test Case 1

also significantly lower below: Indeed, to respect the equilibrium in rotation, the bending moment induced by the corrosion (trough the material softening) must be balanced by a counter-acting bending moment. This is achieved with this decrease in stress. The body exhibits thus the behavior of a plate submitted both tension and flexure. Diagrams of stresses σ_{xx}, σ_{yy} and τ_{xy} may be found in Figure A.4 at page A4 in appendices.

Note that in this model, we simply took into account a damage law reducing material stiffness to model material softening. This law highly related to plastic behavior does not mean that accounted for "real" plasticity! Indeed, the material behavior was still assumed as elastic, with a lower Young modulus and hence, a fully corroded element, can potentially reach values of stresses higher than the yield strength of non damaged material, provided that it is sufficiently deformed. A such case is of course not admissible if plasticity was taken into account. These considerations may be illustrated by the steady-state solution of the problem: once the corrosion is entirely diffused inside the material, the Young's modulus is uniformly distributed all over the plate. The code will provide a solution[2] involving values of horizontal displacements much higher than the actual plastic capabilities of the material.

The numerical model has also been tried on a irregular mesh. For a coarser mesh, results obtained in term of displacements an corrosion propagation are trustful to the results presented in this section. The stresses were however a few less accurate especially in zones characterized by meshes disturbances, even if the global idea was pretty well represented. Results are displayed in appendices in

[2]Excepted in the particular case of $E_d = 0$ i.e. the deteriorated material has no stiffness.

Parameter	Value	Unit
$c(x, y, 0)$	0	%
$c(0, 0.25, t)$	153.8	%
c_{cr}	100	%
D	0.1	$\mathrm{m\,s^{-2}}$
Δt	0.03	s
`MaxTime`	3	s
E_0	200000	MPa
E_d	$0.01E_0$	MPa
M	0	$\mathrm{MPa^{-1}}$
`ElemType`	`Quad4`	-

Table 4.2: Parameters used for test case 1

Figure A.5.

This first test case has proven the ability to model the influence of the corrosion to the mechanical behavior. The material softening induced by the corrosion leads to a plastic redistribution all around the severely corroded zones, and is properly represented by the implemented numerical model.

4.6.2 Test Case 2

The second test is the complement of the first: it will examine the other coupling i.e. how does the model represent the influence of the mechanical loading on the corrosion phenomenon. In order to focus on this aspect only and discard of any parasitic phenomenon inherent to the material degradation, it is proposed to assume that the corrosion rate c does not affect the material stiffness. Therefore, it is assumed that $\mathbf{E}_d = \mathbf{E}_0$ for this second test case.

The test case consists in a plate submitted to a vertical pressure at the right part of its top border. This plate is supported on its lower border. A constant corrosion rate $c = 153.8\%$ is applied at the center, all along the plate height. A such boundary condition is of course not really physical since the corrosion is expected to diffuse from the exterior world (i.e. a boundary), but it makes sense as we try to model the influence of the mechanical behavior on the corrosion diffusion, which require a situation involving high gradients of pressure. The situation is schematically summarized in Figure 4.9 and the parameters used for this simulation are shown in Table 4.3.

Observations

The numerical predictions are shown in Figure 4.10, where the corrosion distribution is represented for 4 values of time.

At time $t = 0$ s, the distribution is perfectly symmetrical (enforced by the boundary conditions). As time goes by, we deviate incontestably from a the symmetrical distribution that the natural diffusion would have exhibited. A pref-

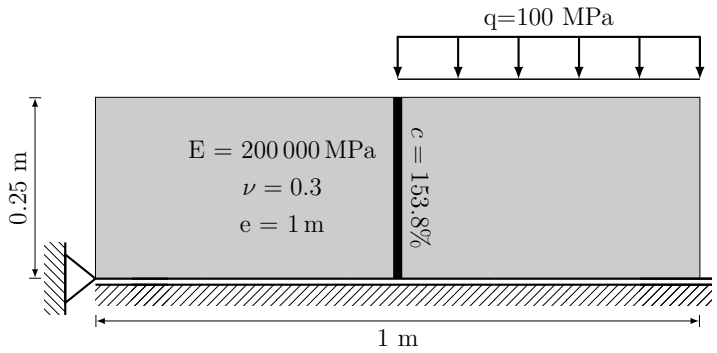


Figure 4.9: Description of test case 2.

erential diffusion direction is clearly observed as the corrosion front progresses faster to the left than to the right part of the plate. To a lesser degree, the diffusion speed is also more important at the top of the plate in its left part, by contrast with the right part where the speed is lower in the corresponding region. As a consequence of this gradient of speed, the propagation front inclines progressively which tends to be more visible with time increasing.

It is also of interest to note that if several zones are characterized by severe (and "preferential") corrosion, it cannot be ignored that corrosion c has evolved everywhere, and not only in the preferential zones. For instance, in Figure 4.10(c), the minimum corrosion rate raises at 25% approximately.

Results discussion

To corroborate and understand the numerical results, let us come back to equation (4.1). This one states that the time variation in c is influenced by both a purely diffusive flux and a non-fickian flux. The first flux is completely independent of the stress state and models a diffusion identical to the one described in Chapter 2, while the non fickian-flux depends on the corrosion rate c, the pressure p, and has a amplitude modeled by the pressure factor M. Hence, the corrosion time evolution is governed by two sources. Based on this, we can explain the two main observations previously made on Figure 4.10.

The first on is about the overall propagation of the corrosion, and is absolutely arising from the natural diffusion term which cause to corrosion to diffuse horizontally and indifferently from the stress state.

The reasons of the preferential directions of diffusion may be found in the second term, the non-fickian flux, which may be written as follow, provided that D and M are constant:

$$\text{Non-Fickian flux} = -\nabla \cdot (DMc\nabla p) = -DM \left(\underbrace{\nabla c \cdot \nabla p}_{=\dagger} + \Delta p \right) \tag{4.56}$$

Before we go any further, let us consider the hydrostatic pressure distribution[3] shown in Figure 4.11. The analysis of this diagram reveals three remarkable zones:

[3]Note that since the material properties are not influenced by the corrosion, the distribution is constant throughout the time).

the first one is straight vertical and is located at the half the plate. It splits the body into two parts: the right one, which is mostly solicited by the imposed pressure and the left one where the pressure is much less, but still positive due to the stress diffusion pattern. The top of this line is bordered by two concentrated zones where the pressure increases very locally[4]. Therefore, the plate center is characterized by a high gradient of pressure in absolute value, which increases progressively from bottom to top with a steeper augmentation close to the upper border.

Also, the boundary conditions imposed on c a such that immediately left to the plate center, $\frac{\partial c}{\partial x} > 0$ while to its right, $\frac{\partial c}{\partial x} < 0$. And by inspection with Figure 4.11, the gradient of stress is highly negative close to the center. Hence, it follows that

$$\begin{cases} \dfrac{\partial c}{\partial x} \cdot \dfrac{\partial p}{\partial x} < 0 & \text{(left)} \quad (4.57a) \\ \dfrac{\partial c}{\partial x} \cdot \dfrac{\partial p}{\partial x} > 0 & \text{(right)} \quad (4.57b) \end{cases}$$

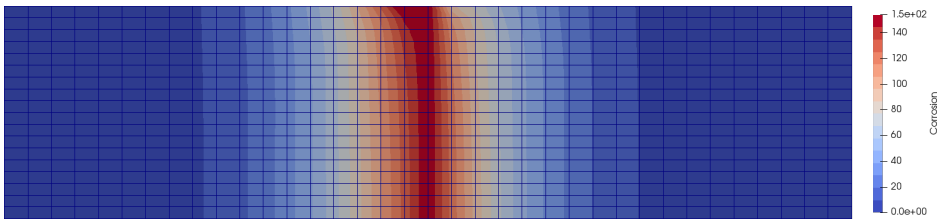
Substituting these two equations into (4.56) and assuming that the term Δp is low, the non-fickian flux is positive in the left direction and negative in the right direction.

Therefore, the time variation in c which is the sum of the two sources contributing to the diffusion process will be lower in the right part than in the left one. The module of the gradient of pressure determines the magnitude of the term † of (4.56). As $\frac{\partial p}{\partial x}$ increases as we get closer to the top, the non-fickian flux will have even more impact at the top of the plate. This is why a accelerated corrosion is observed on the left, and a slower corrosion on the right.

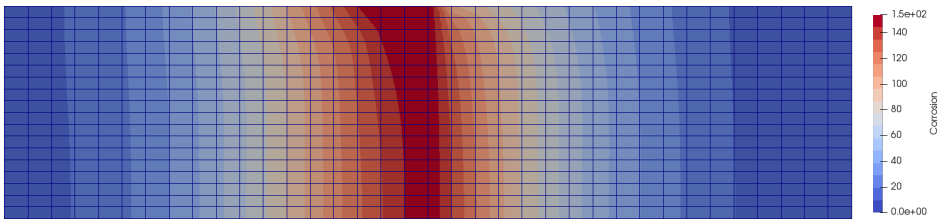
Beyond these considerations, the pressure laplacian Δp adds also a contribution. It takes high positive values when evaluated at a local pressure minimum, and low negative values close to maximums. Initially, the points located at half plate are rather far from the pressure *extrema*, and the assumption of a small laplacian was therefore certainly fair. However, at immediate proximity of the pressure maximum (resp. a minimum), it is sure that the pressure laplacian takes an important role accelerating (resp. decelerating) the diffusion process via the non-fickian flux. It is therefore difficult to identify which term between $\nabla c \cdot \nabla p$ and Δp are responsible to the accelerated corrosion, but the truth lies surely in a combination of both actions.

A interesting observation can also be made in Figure 4.10: the corrosion rate raises in some places values higher than the source corrosion rate (Figure A.7 illustrate the phenomenon without legend trimming). It is difficult to identify the mechanism that makes that the corrosion increases in a such extend, but this one is surely set on by the non-fickian flux as it vanishes for lower pressure factor, or lower applied load.

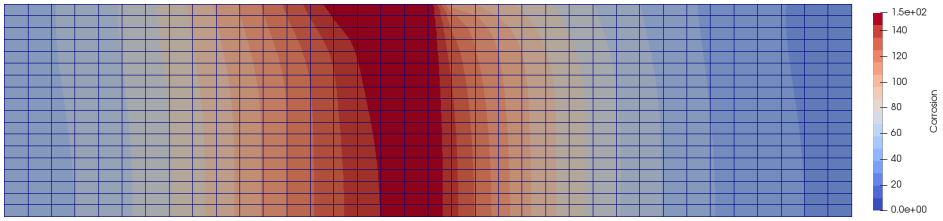
[4]This is indirectly induced by the asymmetrical loading conditions (which generates positive/negative horizontal stresses in these regions)



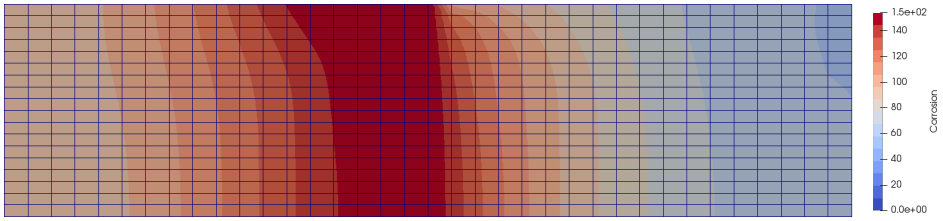
(a) $t = 0.09$ [s]



(b) $t = 0.3$ [s]



(c) $t = 0.6$ [s]



(d) $t = 1.8$ [s]

Figure 4.10: Time evolution of corrosion rate for Test Case 2 (Quad4 Elements)

Conclusions

This objective of this test case was to assess the ability of the developed code to model the a stress-driven diffusion phenomenon. The corrosion has experienced a diffusion into a well determined preferential direction, even if each solid particle has raised its corrosion rate. A comparison with the governing equation (4.1) has corroborated the main features represented in the numerical predictions.

The test case 2 has shown that the corrosion diffuses faster (resp. slower) in zones characterized by pressure maximums (resp. minimums). An accelerated (resp. decelerated) diffusion was also observed toward directions where the corrosion decreases (resp. increased) with the pressure. One main common observation is that the diffusion took place faster in zones where the material was subjected to tension.

Parameter	Value	Unit
$c(x, y, 0)$	0	%
$c(0.5, y, t)$	153.8	%
c_{cr}	100	%
D	0.1	$\mathrm{m\,s^{-2}}$
Δt	0.03	s
`MaxTime`	3	s
E_0	200000	MPa
E_d	E_0	MPa
M	0.3	$\mathrm{MPa^{-1}}$
`ElemType`	`Quad4/Quad9`	-

Table 4.3: Parameters used for test case 2

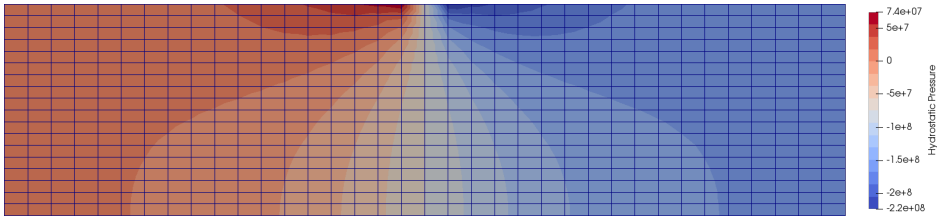


Figure 4.11: Hydrostatic pressure distribution (Quad4 Elements)

4.6.3 Test Case 3

For this last test case, it is proposed to consider the full coupling system: then, both the influence of the corrosion on the mechanical behavior and vice-versa will be modeled. It consists in a cantilever plate, subjected to a constant transverse loading applied on its lower border, as shown in Figure 4.12.

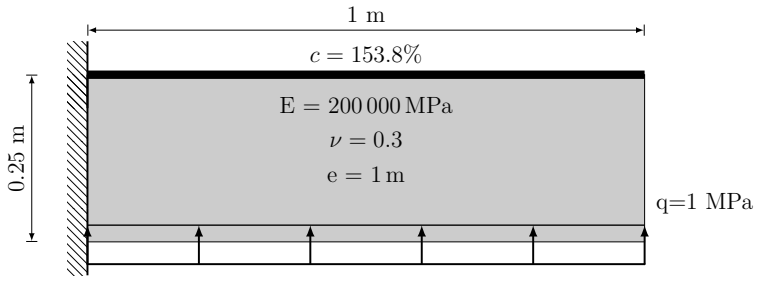
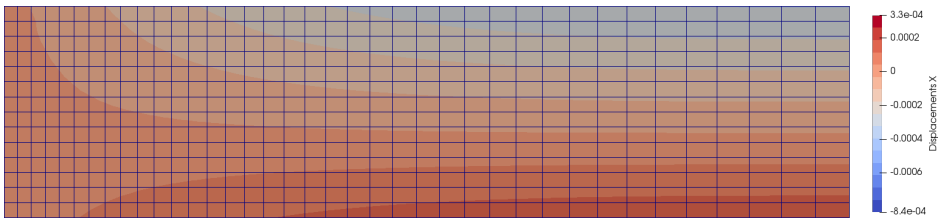


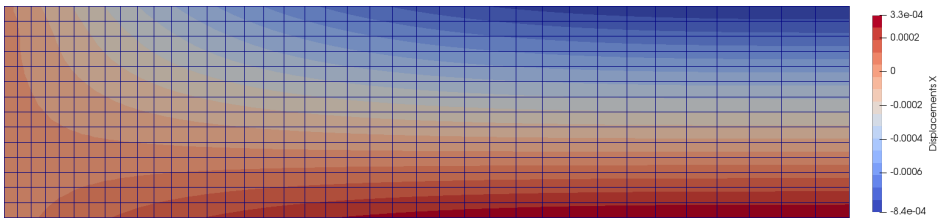
Figure 4.12: Description of test case 3.

Figure 4.13 presents the time evolution of the displacements. The plate behaves initially as a common cantilever plate, but the high corrosion rate imposed at the top causes the neutral axis to be shifted down. As the corrosion diffuses toward the bottom of the plate, the material is progressively soften and the compression zone is gradually enlarged, by opposition to the tensile zone which becomes more concentrated.

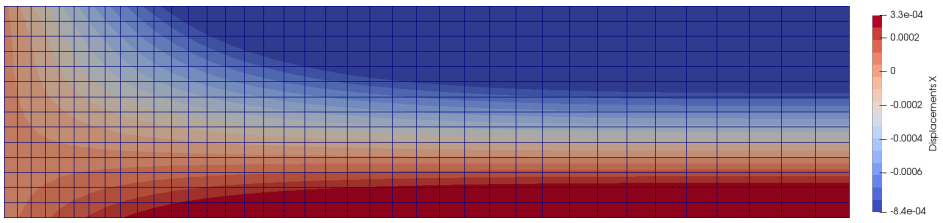
The diagrams of the hydrostatic pressure p are shown in Figure 4.14, for at 3 different time values. It is majorly conditioned by the axial stresses. Universal considerations of mechanics of materials state that the bending moment is



(a) $t = 0.45$ [s]



(b) $t = 1.2$ [s]



(c) $t = 1.8$ [s]

Figure 4.13: Time evolution of horizontal displacements for Test Case 3 (Quad4 Elements)

quadratic and reaches its maximal value $qL^2/2$ at the support, and is zero at the other extremity. Classical beam theory, even if not completely valid for lower aspect ratio plates, tell us that

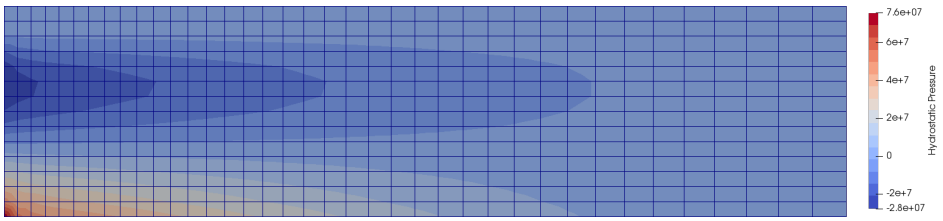
$$\sigma_{xx} = \frac{My}{I} \tag{4.58}$$

This feature is well recovered on each pressure diagram (as σ_x are the dominant stresses), showing two zones in tension or compression separated by a neutral axis gradually pushed down as the corrosion progresses.

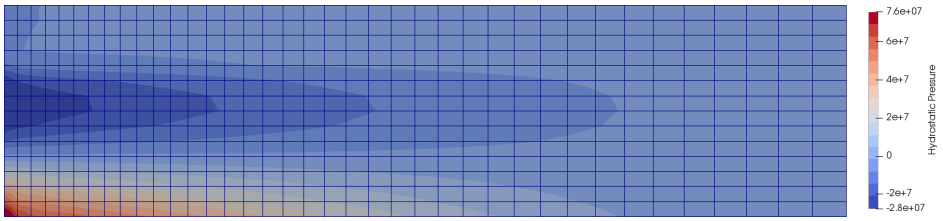
The evolution of the corrosion rate c is shown in Figure 4.15. Initially, the diffusion seems to take place on a natural manner (i.e. the diffusion front is straight): the zones of high stress variation are located lower in the plate and therefore, do not influence the diffusion process. The influence of the minimal pressure (highest compression zone) is visible few after (around 1s) and is perceptible in Figure 4.15(b). A zoom of the corrosion level in the minimal pressure neighborhood is shown in Figure A.6, where it is seen that the corrosion is slower.

Below this zone, the pressure reaches a maximum, and this induces an accelerated corrosion diffusion. Certainly because the pressure laplacian is here negative. Remember from second test case that the directions in which the corrosion rate decreases and in which the pressure increases are favored for accelerated corrosion. These diffusion scheme is well observed in Figure 4.15.

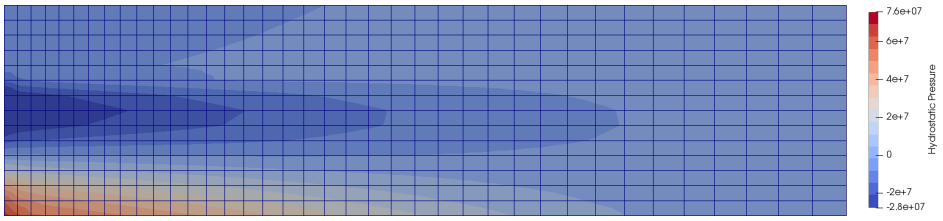
It is of interest to compare the present results with by Figure 4.16 where the same test case was simulated, but with a reversed loading. The pressure diagram,



(a) $t = 0.45$ [s]



(b) $t = 1.2$ [s]



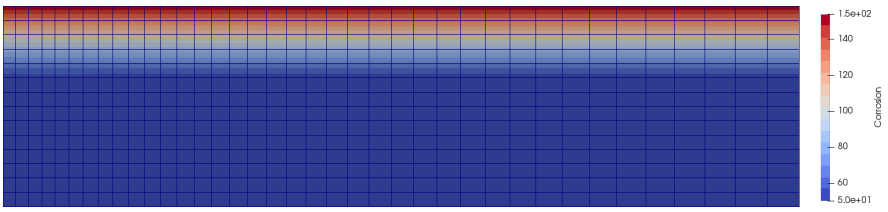
(c) $t = 1.8$ [s]

Figure 4.14: Time evolution of the pressure for test Case 3 (Quad4 Elements)

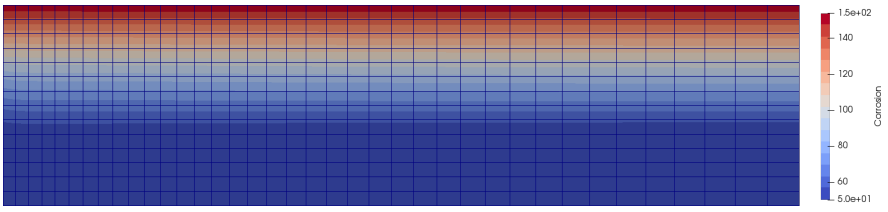
combined with the enforced boundary conditions suggest that only the right upper part of the plate is likely to experience accelerated corrosion. This is well observed in the corrosion graph showing that the corrosion front is deeper on the right. This front appears also to be lower than what was identified in Figure 4.15, and for good reason because the pressure gradient is much less.

The analysis of Figure 4.12 shows that the corrosion induced damage and the associated stress redistribution results in the accentuation of the pressure peaks, which in turn tends to accelerate the diffusion phenomenon. For comparative purposes, Figure 4.17 shows the corrosion distribution at time $t = 1.5$ s for $E_d = E_0$ (i.e. the corrosion does not induce any damage and thus, stress redistribution). Even if well perceptible, the accelerated corrosion occurs to a smaller extent than in the full coupling model.

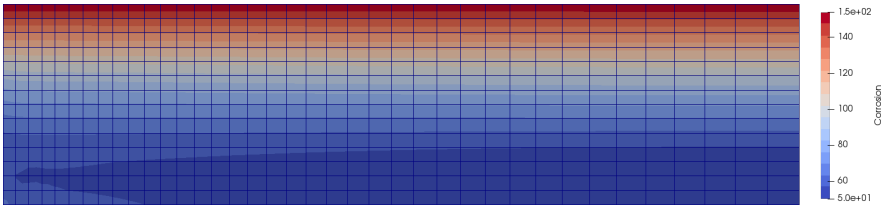
The conclude this test case, we note that the results integrated the main features of the observations made in the two first test cases: the accelerated corrosion and the stress redistribution. In this full coupling model, stress redistribution caused displacement of peak of pressure, but above all, was found to accentuate the maximum of pressure, and subsequently the gradient of pressure. Since pressure gradients are known to influence majorly the non-fluckian flux, this full coupling model has shown faster corrosion propagation due to material damage.



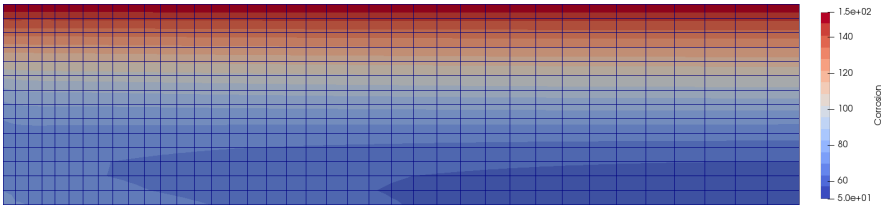
(a) $t = 0.45$ [s]



(b) $t = 1.2$ [s]

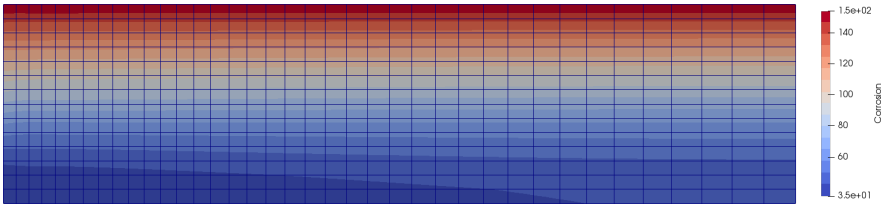


(c) $t = 1.5$ [s]

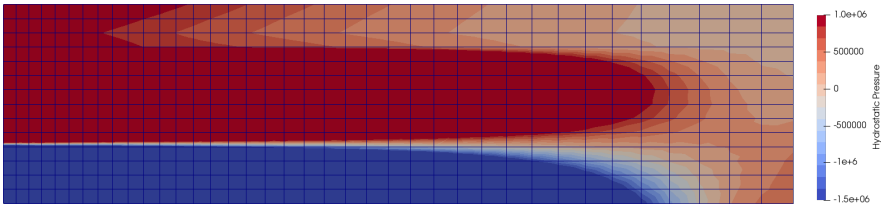


(d) $t = 1.8$ [s]

Figure 4.15: Time evolution of corrosion rate for Test Case 3 (Quad4 Elements)



(a) Corrosion rate



(b) Pressure diagram (legend was trimmed for illustration purposes)

Figure 4.16: Corrosion and pressure diagram at $t = 1.8$ s assuming that the load applied in test case 3 is reversed.

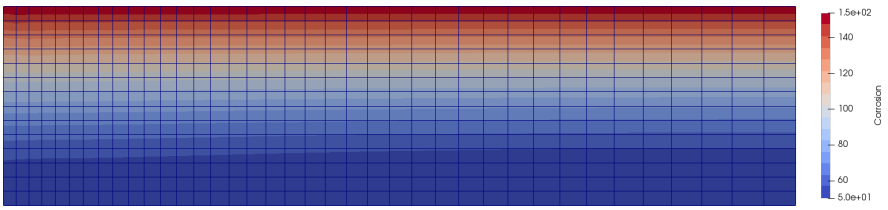
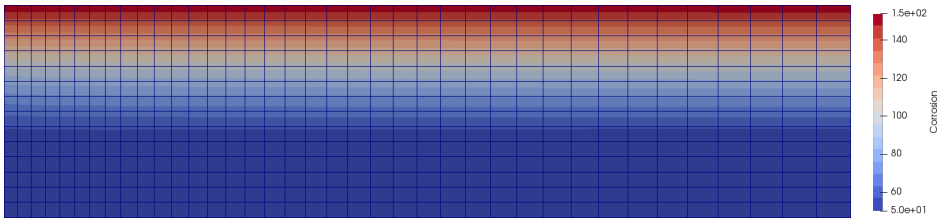
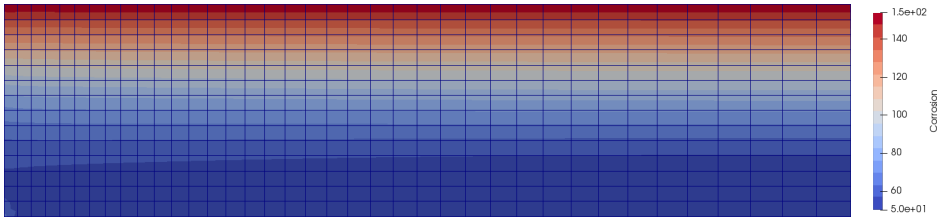


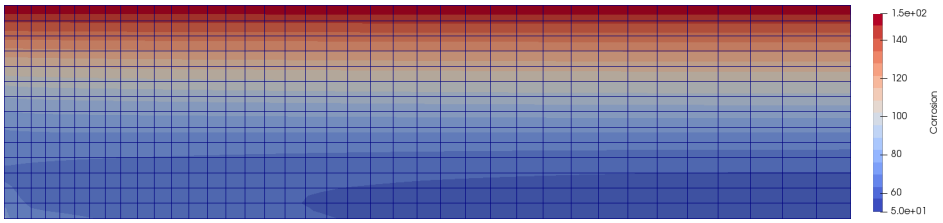
Figure 4.17: Corrosion state at $t = 1.5$ s assuming no corrosion induced damage ($E_0 = E_d$)



(a) $t = 1.2$ [s]



(b) $t = 1.5$ [s]



(c) $t = 1.8$ [s]

Figure 4.18: Time evolution of corrosion rate for Test Case 3 (`Quad9` Elements)

4.6.4 Influence of the element type

One concerning issue mentioned in the establishment of the equations was the ability of the `quad4` elements to represent trustfully a gradient of stresses. To compare the results obtained from 4 and 9-noded elements, the test case 3 has been considered with the same parameters.

Light differences are observed between Figure 4.15 and Figure 4.18. The `quad4` elements predicts a slightly faster diffusion but the diffusion pattern remains however highly similar. The gain in accuracy appears then to be very derisory compared to the dramatic increase in computational cost. For this study case, the calculation time was approximately 20 times slower, for a same number of elements.

As a conclusion, 9-noded elements were found to be unnecessary for this situation. Gradients of stresses in any direction are properly modeled by linear finite elements and therefore the use of quadratic elements is not a necessary condition

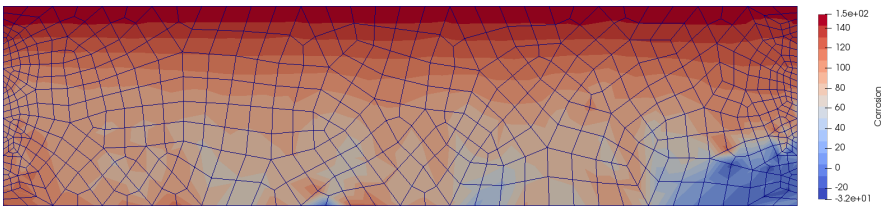


Figure 4.19: Results for Test case 3 with unstructured mesh at $t = 2.04$ s

to meet the requirements of this stress-driven diffusion model.

Nevertheless, it must be noted that 9-noded elements may be helpful to avoid highly refined meshes in critical zones.

4.6.5 Influence of mesh

Up to now, the test cases have only been studied on structured meshes. The code has indeed experienced several difficulties to work with unstructured meshes. In cause, the non-fickian flux which is very sensitive to variation of pressure. The element mismatch orientation inherent to unstructured mesh causes some troubles for the pressure calculation. Firstly, small deviations are punctually observed in several critical points of the mesh. This creates artificial gradients of pressures that influence the corrosion diffusion, which in turn induces a material softening. Stress redistribution occurs around, and the pressure evolves even more: a self-catalyzed diverging process is then triggered. A $\mathbf{u} - p$ mixed-formulation FEM model should be investigated to ensure a $\mathbf{C}_0$ continuity on the pressure field and therefore smoothing the pressure field.

Figure 4.19 shows the results obtained for an unstructured mesh tested on the third test case. Same kind of instabilities have been observed for quad9 elements, but the system started diverging later in the simulation.

This page is intentionally left blank.

Chapter 5

Conclusions and Perspectives

The work aimed at modeling the interaction between corrosion diffusion and mechanical behavior of metallic alloys by the development of a FEM coupling. Several models have been developed in the frame of this master thesis, each integrating an additional difficulty or aspect of the problem. The working process followed during the establishment of all models was based on 4 key steps: the development, the coding, the verification and the validation. All the numerical calculations were carried out in `Fortran 90` to facilitate the potential future model integration with already developed models.

A one-dimensional model for diffusion was firstly developed, assuming that the corrosion was governed by a natural diffusion phenomenon. Even if relatively basic, this first code enabled to build the main structure of the code. Beyond these practical considerations, it initiated the discussion of the stability of the system. The partial time derivative was discretized following an implicit integration scheme, whose main advantage with respect to explicit ones was the absence of stability requirements of any kind. This choice penalized the computational cost.

The verification of this model has been performed based on the analytical solution of the differential equations derived for two particular test cases involving both essential and natural conditions. The evolution of an RMS error has been studied for multiple numbers of finite elements. A convergence was well obtained, since the error was progressively tending to zero. The time evolution of the same error has also been studied, and was marked by a peak of error at the beginning of the simulation. This was due to the brutal changes in concentration imposed by the boundary conditions. This peak vanished immediately once the distribution was smoothed. The obtained errors were entirely dependent on the mesh finesse, and indifferent from the chosen time step.

For this one-dimensional model, the validation step (i.e. the comparison of numerical predictions with experimental results) was performed based on existing literature [1, 3], were experiments have been carried out on hot-temperature oxidation of RR1000 and titanium alloys. Unfortunately, some information were missing to obtain a perfect matching between empirical and numerical results (for example corrosion diffusivity D was not evaluated). A literature research [1,20,21] provided some more or less accurate methods to determine the missing pieces of

information. It was concluded that the code was able to predict the time evolution the oxidation layer depth, for suitable value of diffusivity D. Despite of errors reaching sometimes 25% with respect to the experimental results, a good correlation of the numerical results was always observed, indicating that the natural diffusion equation assumed at the beginning was well adapted to model the diffusion of corrosive components into non-stressed materials.

Secondly, the model has been extended to the two-dimensional space. A post-processing environment has been developed in `Python` to enable visualization of 2D-data in `Paraview`. Similarly to the first model, linear shape functions have been used. In this model however, isoparametric elements have been considered (quad4). For verification purposes, two test cases have been considered to assess the ability of the code to model the diffusion with both essential and natural boundary conditions. These test cases were such that 1D-diffusion only was experienced, and the results were confronted to the 1D analytical solution. The code was verified on this basis, with regular as well as highly irregular meshes. Due to the absence of references for 2D corrosion diffusion, the validation was performed based on physical intuition.

In parallel, the model of mechanical behavior in two dimensions has been developed. Starting from the 3D-equilibrium equations for classical mechanics, the weak form of the problem was derived to recover the virtual work statement. This displacements-based model was build under the assumptions of a linear elastic behavior of solids in plane stress state, with isoparametric linear elements (quad4). Afterwards, a stress recovery procedure including a local smoothing was implemented in the code: Indeed, after resolution of the linear system, the displacements at nodes are known, and the stresses may be accordingly calculated at gauss points, but should never be directly extrapolated at nodes to avoid unrealistic stress jumps troubling the results interpretation. This stress recovery procedure aims to decrease the error of a polynomial recovered stress field over each finite element, in sense of root mean square and discards the irrelevant stress discontinuities at the element interfaces.

The mechanical model was tested on two situations. The first one, a plate in pure extension, served as patch test and passed the validation; the constant strain was properly modeled, matching the analytical solution. The second envisaged test case was a cantilever plate subjected to bending. The results were compared to a trustfully structural analysis software, under the same assumptions of plane stress state and linear shape functions. Displacements were exactly recovered while the stresses values experienced slight acceptable deviations, surely due to the difference in stress recovery procedures.

The last part of the work was dedicated to the coupling of the two previous models. The diffusion process, so far considered as purely diffusive, was enriched by a second non-fickian flux, which models the influences of the stress state on the diffusion phenomenon. Such process was qualified as stress-driven corrosion process. The equation describing the material behavior remained unchanged, as an linear elastic behavior was still assumed. However, the corrosion induced damage was taken into account by means of a linear damage law. This latter law established a relationship between the material Young's modulus and a damage variable ϕ, defined as the minimum between the corrosion rate and 100%: the

closer ϕ to 100%, the lower the Young's modulus. This ends up to each residual stiffness from which the stiffness does not evolve anymore (ϕ has there reached 100%). The existence of this residual stiffness should not be expected in reality, but is necessary to avoid numerical divergence.

The scope of this work has been initially defined in a such way that non-linear analysis will not be performed: it was indeed decided to harness and assess the available possibilities involving the resolution of a linear system. However, it turned out that the coupling system of equations formed by the stress-driven diffusion and the mechanical behavior was non linear. The sources of non-linearities were found in the stress-driven diffusion equation (4.1) itself, but were also incorporated by the expression of the variable Young's modulus. To linearize the formed differential system of equations, two hypothesis were investigated. The first one assumed $\mathbf{q}_c^{t+1} \approx \mathbf{q}_c^t$, and was auto-sufficient to linearize the system. The second option relied on the assumption $\mathbf{q}_u^{t+1} = \mathbf{q}_u^t$. This latter assumption needed to be supplemented by the same hypothesis as done in the previous option. The system was then degenerated into a diagonal system matrix, and thus a staggered coupling model. It was relinquished for the benefit of the first one, which was judged as more consistent.

Regarding the solving of the differential equation system, we preferred an implicit integration scheme rather than an explicit one for its unconditional stability, and also because the explicit schemes have lost their interest now that our stiffness matrix are now dependent on c, which varies in time (and therefore, the iteration matrix must be re-calculated at each iteration). The linearization performed to derive the expression of the first coupling option has weakened the integration scheme stability that we took benefit so far: as the scheme is no more fully implicit, the stability is associated to the time step, which should not be taken too large to avoid high deviations from the tangent stiffness matrix. A deeper study of the approached implicit scheme used should be carried out on its stability. Failing that, a trustful value of Δt was derived for a given situation by decreasing progressively its value until convergence was observed.

The numerical model was finally tested on three situations. The first one aimed to model the corrosion induced damage on the mechanical behavior. It was mainly observed that the material softening induced by corrosion caused a stress redistribution in stiffer zones. Asymmetrical boundary conditions enforced on c have shown to modify the overall behavior of the solid: initially in pure tension, the stresses redistribution caused the apparition of vertical and shear stresses, that take more and more importance as the redistribution takes place.

Plasticity was not taken into account in this work. Instead we assumed a linear elastic behavior, with variable stiffness. This hypothesis may lead to unrealistic results for example when corrosion tends to be locally or globally generalized. In such case, stress redistribution allows unrealistic deformations of the body to satisfy the equilibrium equations, even if the material properties are outrageously violated. A strain limit at material failure, putting an end to the simulation, should at least be considered to avoid unreasonable results.

The second test case studied a stress-driven corrosion phenomenon. The corrosion was found to diffuse faster into directions where the pressure gradient was increasing with a decreasing corrosion rate. Local maximum of positive pressure

were also accelerating the diffusion process. By opposition with zones marked with a peak of compression which was decelerating the process. For high pressure factor ($M > 0.3\text{MPa}^{-1}$), non-fickian flux became prominent on the fickian flux, and corrosion rate higher than the source values was sometimes observed. Further research on this phenomenon should be carried out.

The full coupling model was finally tested on the third test case. The softening induced by the corrosion led to a stress redistribution which increased further the gradient of pressure, displaced the location of the pressure *extrema* and increased maximal pressure values. As a consequence, the corrosion diffusion modeled in this test case was strongly influenced by the stress state. This became even more evident as the corrosion was spreading out and the stress redistribution took place.

It was also proven that the 4-noded quadrangles are suitable for this coupling model, and represent properly a variation of hydrostatic pressure, provided that the mesh is sufficiently fine, especially in zones where the pressure may be expected to vary rapidly. 9-noded elements have shown few improvements of the transient solution, corroborating with light deviation the predictions of the quad4 elements.

The full coupling model experienced a bad behavior with unstructured meshes, diverging when the stress-driven diffusion was starting to influence the corrosion evolution. Further studies should be carried out on a $\mathbf{u} - p$ mixed-formulation model, to determine if a smoothed pressure field could not fix the phenomenon observed. As the targeted applications involve mechanical parts of complex geometries, it is paramount that the code is able to handle unstructured meshes, before to be extended to the 3 dimensional space.

Several additional perspectives can also be considered. For instance, the material swelling inherent to the chemical corrosion reaction could be integrated to the model. A correlation describing the evolution of the diffusivity D as a function of the oxygen rate could also be incorporated.

Bibliography

[1] A Encinas-Oropesa, G.L. Drew, Mark Hardy, A.J. Leggett, John Nicholls, and NJ Simms. Effects of oxidation and hot corrosion in a nickel disc alloy. pages 609–618, 01 2008.

[2] A. Karabela, L.G. Zhao, B. Lin, J. Tong, and M.C. Hardy. Oxygen diffusion and crack growth for a nickel-based superalloy under fatigue-oxidation conditions. *Materials Science and Engineering: A*, 567:46 – 57, 2013.

[3] Dimitris C. Lagoudas, Pavlin Entchev, and Robertus Triharjanto. Modeling of oxidation and its effect on the crack growth resistance of titanium alloys. In George Z. Voyiadjis, Jiann-Wen Woody Ju, and Jean-Louis Chaboche, editors, *Damage Mechanics in Engineering Materials*, volume 46 of *Studies in Applied Mechanics*, pages 421 – 440. Elsevier, 1998.

[4] Pôle Mecatech. Solar perform. `https://www.polemecatech.be/fr/projets/solar-perform/`, 2016-2019.

[5] Pôle Mecatech. Solar gnext. `https://www.polemecatech.be/fr/projets/solar-gnext/`, 2019-2022.

[6] University of Liège. Lagamine. `https://www2.uee.uliege.be/cms/c_2383455/nl/lagamine`.

[7] Alain FERRIÈRE. Centrales solaires thermodynamiques. *Techniques de l'ingénieur Énergies renouvelables*, base documentaire : TIB594DUO.(ref. article : be8903), 2008.

[8] Sandia National Laboratories. Final test and evaluation results from the solar two project, 2012.

[9] Frederick Redell (Abengoa Solar USA) . Solar thermal electric plants: Enabling a reliable grid. `https://www.solarnovus.com/solar-thermal-electric-plants-enabling-a-reliable-grid_N9183.html`, 2015.

[10] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[11] Ahrens, James, Geveci, Berk, Law, and Charles. *ParaView: An End-User Tool for Large Data Visualization*, 2005.

[12] O.C. Zienkiewicz and R.L. Taylor. *Discrete approximation in time*, volume 2. 4 edition, 1991.

[13] Quentin Louveaux. *Introduction to Numerical Analysis*. University of Liège, Montefiore Institute, 2015.

[14] Richard L. Burdean, J. Douglas Faires. *Numerical Analysis*. Brooks/Cole Cengage Learning, 9th edition, 2010.

[15] Richard L. Burdean, J. Douglas Faires. *Numerical Analysis*, chapter Numerical Differentiation and Integration, pages 228–232. Brooks/Cole Cengage Learning, 9th edition, 2010.

[16] Singiresu S Rao and Yap Fook Fah. Continuous systems. In *Mechanical vibrations in SI Units*, chapter 8. Prentice Hall, 5th edition, 2011.

[17] Frank P. Incropera. *Fundamentals of Heat and Mass Transfer*. John Wiley & Sons, Inc., USA, 2006.

[18] M.A. Rocazella and I.G. Wright. The corrosion behaviour of some candidate heat-exchanger alloys in the afbc environment. *Proceedings of the Symposium on High Temperature Materials Chemistry*, 98:408–419, 1998.

[19] David J. Young. Chapter 2 - enabling theory. In David J. Young, editor, *High Temperature Oxidation and Corrosion of Metals (Second Edition)*, pages 31 – 84. Elsevier, second edition edition, 2016.

[20] Hammad Nazir, Zulfiqar Khan, Adil Saeed, and Keith Stokes. Modelling the effect of residual and diffusion induced stresses on corrosion at the interface of coating and substrate. *Corrosion*, 72:500–517, 12 2015.

[21] Z. Azari, M. Abbadi, H. Moustabchir, and M. Lebienvenu. The influence of fatigue cycling on the oxidation kinetics and crack initiation of a cr–mo steel. *International Journal of Fatigue*, 30(3):517 – 527, 2008.

[22] Yaohong Suo, Zhang Zhonghua, and Xiaoxiang Yang. Perturbation approach to metal surface oxidation considering volume variation. *Advances in Mechanical Engineering*, 9:168781401667678, 01 2017.

[23] Dhananjay Pradhan, f Girija Mahobia, Kausik Chattopadhyay, and Vakil Singh. Effect of surface roughness on corrosion behavior of the superalloy in718 in simulated marine environment. *Journal of Alloys and Compounds*, 740, 04 2018.

[24] César Sequeira. *High Temperature Corrosion: Fundamentals and Engineering*. 01 2019.

[25] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 5 - mapped elements and numerical integration — 'infinite' and 'singularity elements'. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method Set (Sixth Edition)*, pages 138 – 186. Butterworth-Heinemann, Oxford, sixth edition edition, 2005.

[26] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 2 - a direct physical approach to problems in elasticity: Plane stress. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method Set (Sixth Edition)*, pages 19 – 53. Butterworth-Heinemann, Oxford, sixth edition edition, 2005.

[27] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 3 - generalization of the finite element concepts. galerkin-weighted residual and variational approaches. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method Set (Sixth Edition)*, pages 54 – 102. Butterworth-Heinemann, Oxford, sixth edition edition, 2005.

[28] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 9 - mixed formulation and constraints: Complete field methods. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, pages 285 – 314. Butterworth-Heinemann, Oxford, seventh edition edition, 2013.

[29] Dubravka Mijuca, Z Draskovic, and M Berkovic. Displacement based continuous stress recovery procedure. pages 127–134, 05 2019.

[30] E. Hinton and J. S. Campbell. Local and global smoothing of discontinuous finite element functions using a least squares method. *International Journal for Numerical Methods in Engineering*, 8(3):461–480, 1974.

[31] A.A. Rogovoy. The stress recovery procedure for the finite element method. *Computers & Structures*, 63(6):1121 – 1137, 1997.

[32] J. T. Oden and H. J. Brauchli. On the calculation of consistent stress distributions in finite element approximations. *International Journal for Numerical Methods in Engineering*, 3(3):317–325, 1971.

[33] E. Hinton, F. C. Scott, and R. E. Ricketts. Local least squares stress smoothing for parabolic isoparametric elements. *International Journal for Numerical Methods in Engineering*, 9(1):235–238, 1975.

[34] Josselin DELMAS. Calcul des contraintes aux noeuds par lissage local. *Code Aster (Manuel de référence)*, 2011.

[35] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 8 - the patch test, reduced integration, and nonconforming elements. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, pages 257 – 284. Butterworth-Heinemann, Oxford, seventh edition edition, 2013.

[36] J.M. Frederiksen ; L. Mejlbro ; L.-O. Nilsson. Fick's second law - complete solution for chloride ingress into concrete, 2008.

[37] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 4 - 'standard' and 'hierarchical' element shape functions: Some general families of c0 continuity. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method Set (Sixth Edition)*, pages 103 – 137. Butterworth-Heinemann, Oxford, sixth edition edition, 2005.

[38] O.C. Zienkiewicz and R.L. Taylor. *Finite element of an elastic continuum-displacement approach*, volume 1. McGraw-Hill, 1994.

[39] Yo Tomota, Shin Daikuhara, Shun Nagayama, Masanori Sugawara, Norihiko Ozawa, Yoshitaka Adachi, Stefanus Harjo, and Shigeo Hattori. Stress corrosion cracking behavior at inconel and low alloy steel weld interfaces. *Metallurgical and Materials Transactions A*, 45, 12 2014.

[40] N. Eliaz, G. Shemesh, and R.M. Latanision. Hot corrosion in gas turbine components. *Engineering Failure Analysis*, 9(1):31 – 43, 2002.

[41] Patrice HOULLE. Résistance à la corrosion aqueuse des alliages de nickel. *Techniques de l'ingénieur Métaux et alliages non ferreux*, base documentaire : TIB357DUO.(ref. article : cor312), 2015. fre.

[42] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 6 - problems in linear elasticity. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method Set (Sixth Edition)*, pages 187 – 228. Butterworth-Heinemann, Oxford, sixth edition edition, 2005.

[43] Walter Gautschi. *Numerical Analysis*. Springer, 2012.

[44] Serge Cescotto. *Elements de mecanique des solides deformables*. Liège : Université de Liège Faculté des Sciences Appliquées, 1986.

[45] Paolo Maria Mariano, Luciano Galano. *Fundamentals of the Mechanics of Solids*. Birkhäuser, 2015.

[46] J.P. Ponthot. Chapter 11 - isoparametric elements. Lectures University of Liège, 2016.

[47] Gaëtan Wautelet. Workshop isoparametric elements. Slides University of Liège, 2016.

[48] Suvranu De. Fem discretization of 2d elasticity. `https://homepages.rpi.edu/~des/FEM2&3D.pdf`. Rensselaer Polytechnic Institute.

[49] Piaras Kelly. Differential equations for solid mechanics. `http://homepages.engineering.auckland.ac.nz/~pkel015/SolidMechanicsBooks/Part_II/01_DifferentialEquilibriumAndCompatibility/01_DifferentialEquations_Complete.pdf`, 2017. Univeristy of Auckland.

[50] O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu. Chapter 15 - errors, recovery processes, and error estimates. In O.C. Zienkiewicz, R.L. Taylor, and J.Z. Zhu, editors, *The Finite Element Method: its Basis and Fundamentals (Seventh Edition)*, pages 493 – 543. Butterworth-Heinemann, Oxford, seventh edition edition, 2013.

[51] B. J. Foss, M. C. Hardy, D. J. Child, D. S. McPhail, and B. A. Shollock. Oxidation of a commercial nickel-based superalloy under static loading. *JOM*, 66(12):2516–2524, Dec 2014.

[52] Madjid Sarvghad Moghaddam. Corrosion of structural alloys in molten salts for solar thermal energy storage, 2018.

Appendix A

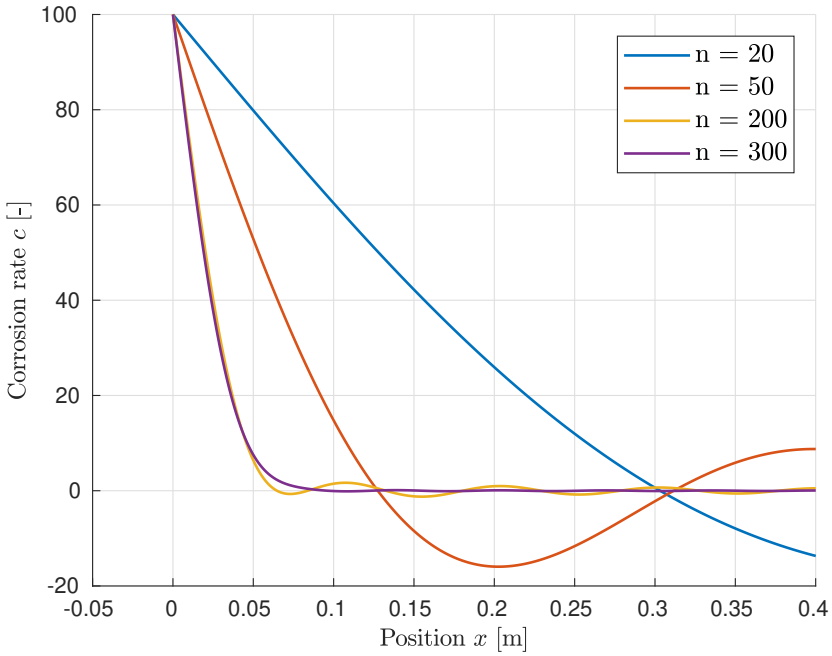
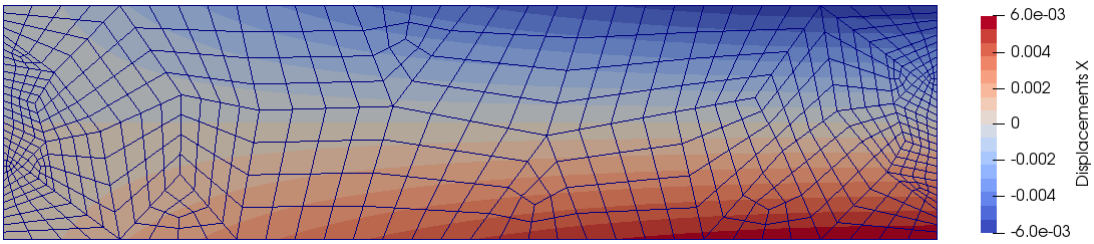
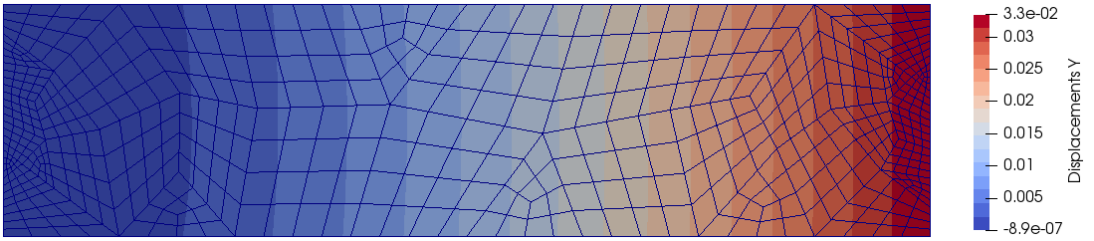


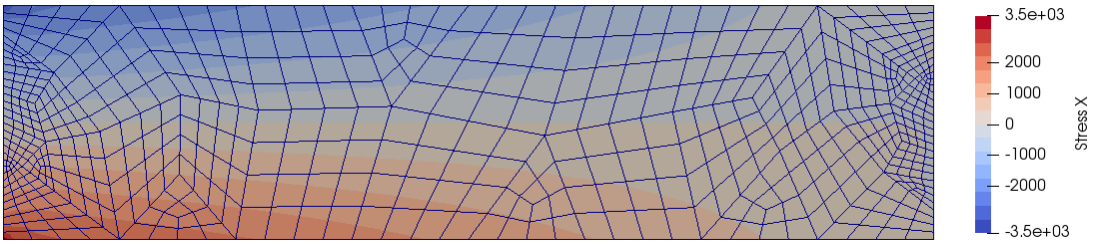
Figure A.1: Illustration of Gibbs phenomenon at time $t = 0.004$ s, and for different truncation of the Fourier series (parameters used: $c_{x_0} = 100\%$, $c_{x_L} = 25\%$ and $c_0 = 0\%$ and $D = 0.1\,\mathrm{m^2\,s^{-1}}$)



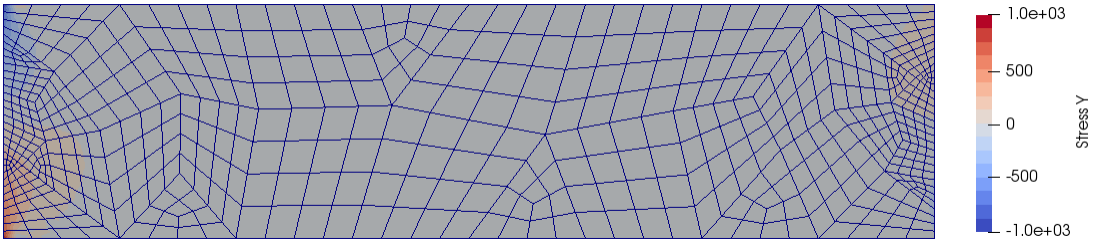
(a) Displacements u [m]



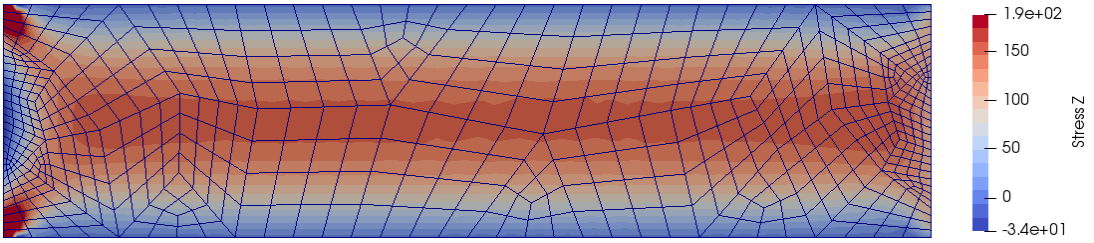
(b) Displacements v [m]



(c) Stresses σ_{xx} [MPa]

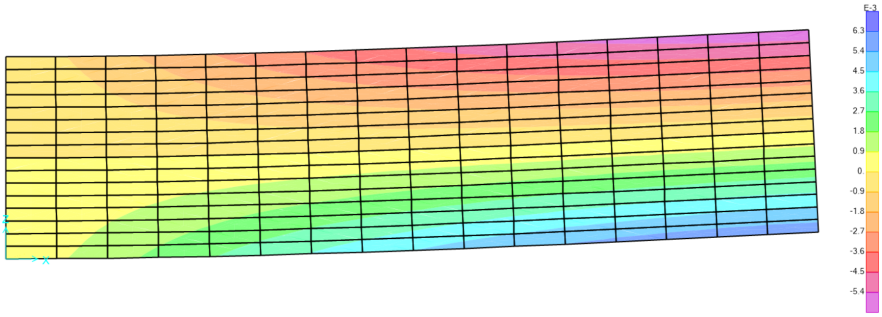


(d) Stresses σ_{yy} [MPa]

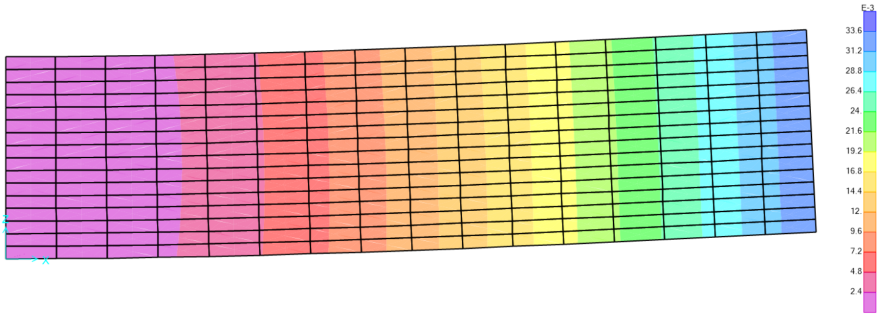


(e) Stresses τ_{xy} [MPa]

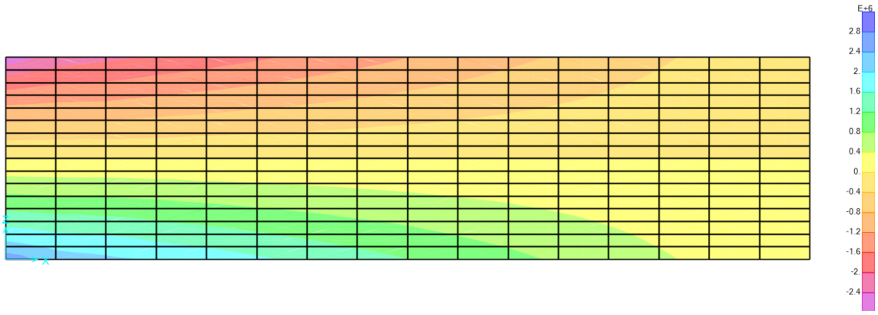
Figure A.2: Results for test case 2 with second order polynomial shape functions



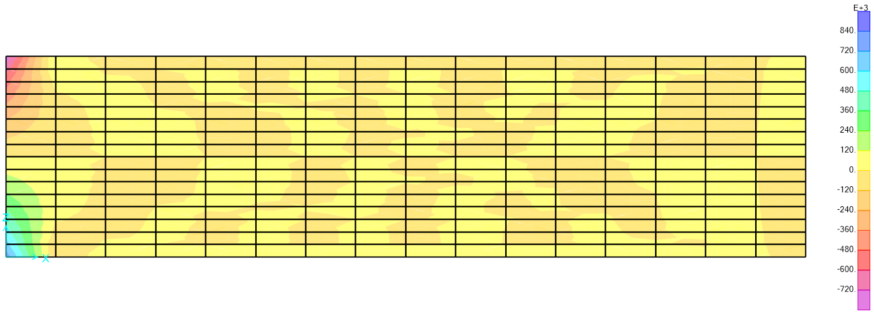
(a) Displacements u [m]



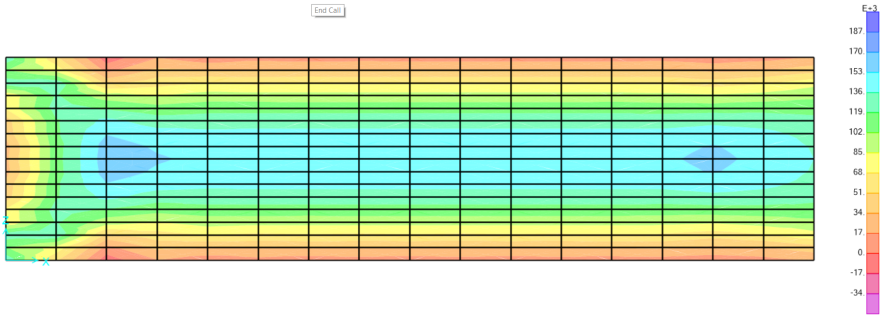
(b) Displacements v [m]



(c) Stresses σ_{xx} [kPa]

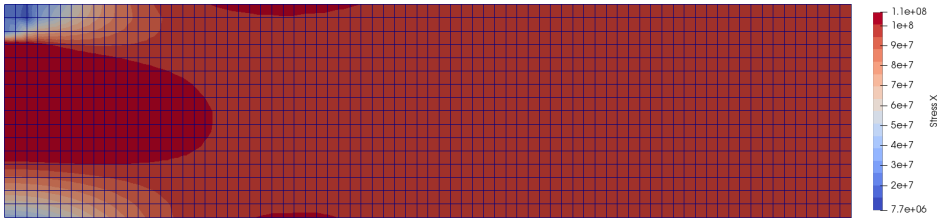


(d) Stresses σ_{yy} [kPa]

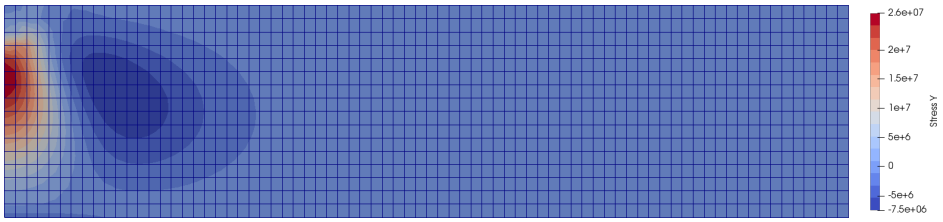


(e) Stresses τ_{xy} [kPa]

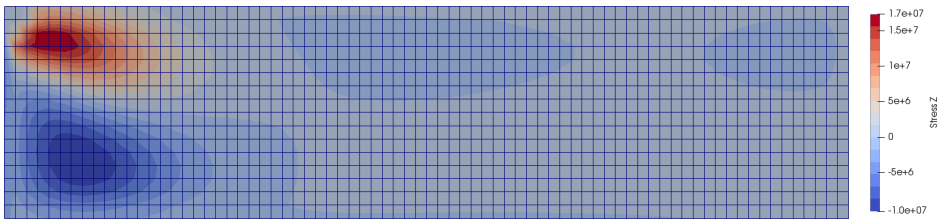
Figure A.3: Results for test case 2, obtained from SAP2000 simulation.



(a) σ_x [Pa]

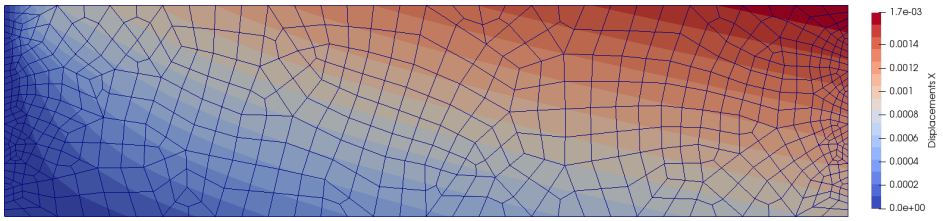


(b) σ_y [Pa]

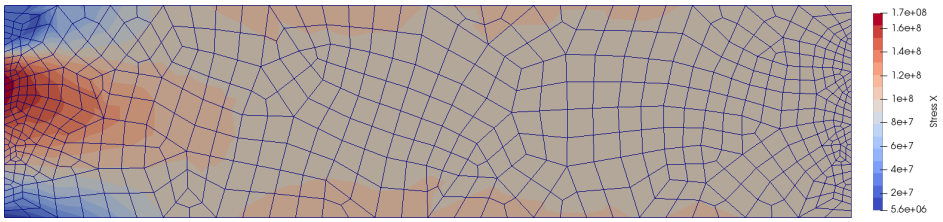


(c) τ_{xy} [Pa]

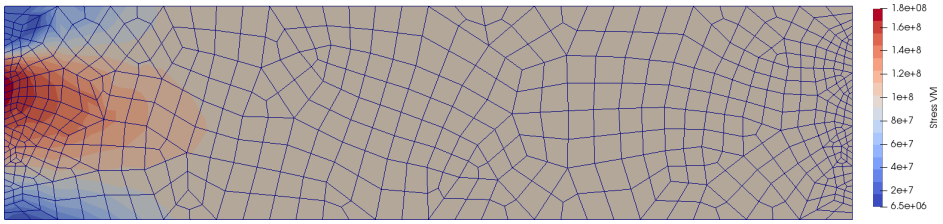
Figure A.4: Stresses obtained for Test Case 1 at time $t = 3\,\mathrm{s}$



(a) u[m]



(b) σ_x [Pa]



(c) σ_{VM} [Pa]

Figure A.5: Several results for first Test Case 1 at time $t = 3\,\mathrm{s}$, for an irregular mesh

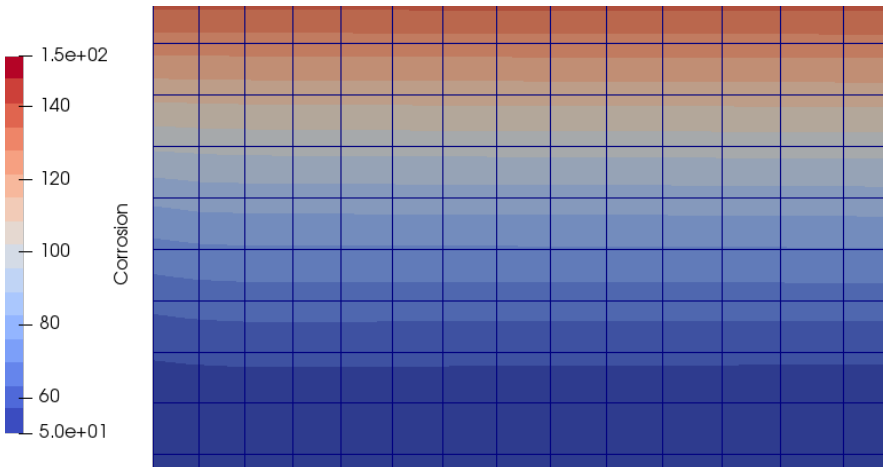


Figure A.6: Zoom on the decelerated corrosion experienced in test case 3

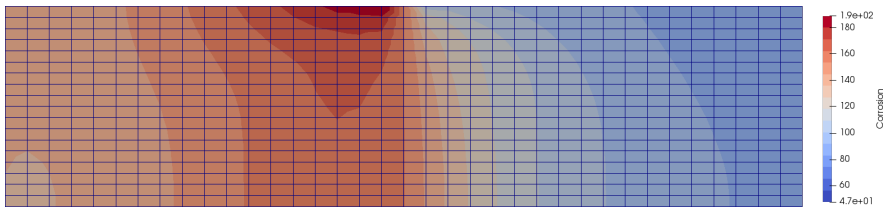


Figure A.7: Corrosion rate is higher than the source value in several locations of the domain