Max Planck Institut for Plasma Physics

Technical Intership

Implementation of a Full-Wave code for the numerical simulation of Plasma-Wave interaction, using a B-Splines Finite Elements method

Author:
Margaux Bruliard

Supervisor:
Dr. Ing. Ahmed Ratnani

in the

Numerical Methods in Plasma Physics

September 26, 2018

Acknowledgements

I would like to take this opportunity to thank all the people who have contributed in some way to this report particularly Mr. Ahmed RATNANI, my supervisor for this internship who give me the opportunity to work with him.

My thanks also go to Mr. Eric SONNENDRÜCKER, the head of the NMPP team, to having welcomed me into his team.

I also thank all the member of the secretariat team of NMPP, Mrs. Anneliese GRASMANN and Mrs. Laura FAHRNER for their availability and their enthusiasm

Finally, I would like to express my deep sense of gratitude for Mr. Olivier LAFITTE, for having recommended me this internship and helping me in the administrative steps.

Contents

List of Figures

Introduction

General Introduction

In the 19th century, the mechanic of Newton and its developments by contemporaries establishes the " rational mechanics " which becomes a model of the sciences of nature. The electricity and the magnetism are science subjects younger than the mechanic and have difficulties acquiring a systematic structure of thought. Indeed these subjects contain difficult questions such as the multiplicity of the actions at distances.

We remember some scientists, precursors of the electricity and magnetism domain: H.C. OERSTED (on 1791 - 1851) who discovers that an electric current in movement produces magnetic effects; M. FARADAY (on 1791 - 1867) who discovers that the movement of a magnet generates electrical currents. These phenomena prove that the electric and magnetic strengths interact between them. It is James Clerk MAXWELL (on 1831 - 1879) who publishes a first work allowing to express useful known laws for the electromagnetism based on the work of several scientists (for example M. FARADAY)[1]

The system of equations presented by MAXWELL in these works and generally called " Maxwell's equations " help to describe the evolution of the system of the electric and magnetic fields. As we know that these fields are the main characteristics of an electromagnetic wave, which are used in numerous domains today such as the radio, the X-rays or the γ beams for example, then Maxwell's equations are today one of the best ways to describe the evolution of an electromagnetic wave.

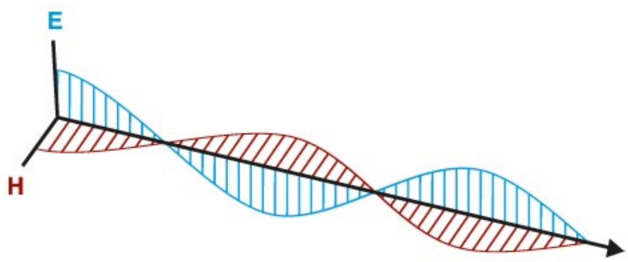

FIGURE 1: An electromagnetic wave representation based on electric and magnetic field

In this report, our aim is to predict the evolution of an electromagnetic wave, and

[1] These details come from the article of Sanchez-Palencia, 2013

more particularly the evolution of the electric and magnetic fields which characterize this wave. In this purpose, we try to solve Maxwell's equations by the finite element method by approximating the solutions by the method of B-splines.

Max-Planck Institute for Plasma Physics

The Max-Planck Institute for Plasma Physics (IPP) is a physics institute based on Germany. This is an institute of the **Max-Planck Society** and part of the **European Atomic Energy Community**. There are two sites: the first one in Garching (near München) founded in 1960 and the second in Greifswald founded 1994. The main research subject of the IPP is the physical foundations of a fusion power plant.

There are 11 scientific divisions at the Max-Planck Institute of Plasma Physics of which **the Numerical Methods in Plasma Physics (NMPP)** division managed by Dr. Eric SONNENDRÜCKER. The purpose of this division is to develop new computational methods with large quantities of data

The MagnetoHydroDynamics working group of the NMPP managed by Ahmed RATNANI works on the study, development, and analysis of robust and efficient algorithms for numerical simulations of different MagnetoHydroDynamics models. The main numerical methods this group is working on are:

- Isoparametric and Isogeometric Analaysis
- High Order Discontinuous Galerki Method
- Mesh Generation
- Compatible Finite Element Methods

This last one subject allows the construction of codes and numerical methods for different functional spaces. This is where the SPL module for Python is developing

SPL module for Python

The SPL module for Python is a part of the **pyceel Python3 library** computing by the MagnetoHydroDynamics working group of NMPP. One of the aims of this open source library is the implementation of finite element methods using B-Splines in a python or Fortran language with the possibility to use parallel computation. The MagnetoHydroDynamics is still working on this project and I really hope my contribution will help them to progess this Python3 library.

Chapter 1

Bézier polynomials

1.1 Bernstein basis polynomials

We call a **Bernstein polynomial** a polynomial with a Bernstein form which is a linear combinaison of Bernstein basis polynomials. We also said the Bernstein polynomials allow a constructive and probability approach of **Weierstrass theorem**.

We consider $t \in [0,1]$, and we can define for every order $p \geq 0$ p+1 Berstein basis polynomials, which can note as $B_0^p, B_1^p, ..., B_p^p$:

$$B_i^p(t) = \binom{p}{i} t^i (1-t)^{p-i} \tag{1.1}$$

where $C_p^i = \binom{p}{i}$ are binomial coefficients.

Property 1.1.1. Directly from (1.1) we have these properties:

1. unity partition: $\sum_{i=0}^{p} B_i^p(t) = 1$
2. positivity: $\forall i \in \{0, ..., p\}, \quad B_i^p(t) \geq 0$
3. symetry: $\forall i \in \{0, ..., p\}, \quad B_i^p(t) = B_{p-1}^p(1-t)$

Proof. 1. **unity partition**:

$$\sum_{i=0}^{p} B_i^p(t) = C_p^i t^i (1-t)^{p-i}$$

and we know by using **binomial theorem** $(x+y)^n = \sum_{k=0}^{n} C_n^k x^k y^{n-k}$, so:

$$\begin{aligned}\sum_{i=0}^{p} B_i^p(t) &= C_p^i t^i (1-t)^{p-i} \\ &= (t + (1-t))^p \\ &= 1^p \\ &= 1\end{aligned}$$

2. **positivity**:

- $\forall t \in [0,1], t \geq 0$ and $1 - t \geq 0$

   - binomial coefficients are positive

   So $B_i^p(t) \geq 0$

3. **symetry**:

   we choose $i \in \{0, ..., p\}$, and we know that $C_p^{p-i} = C_p^i$

$$\begin{aligned} B_{p-i}^p(t) &= C_p^{p-i} t^{p-i} (1-t)^i \\ &= C_p^i t^{p-i} (1-t)^i \end{aligned}$$

$$\begin{aligned} B_{p-i}^p(1-t) &= C_p^i (1-t)^{p-i} t^i \\ &= B_i^p(t) \end{aligned}$$

□

Property 1.1.2. We can also define a recurence formula from (1.1)

$$\forall p \geq 1 \quad B_i^p(t) = \begin{cases} (1-t)B_i^{p-1} & i = 0 \\ (1-t)B_i^{p-1}(t) + tB_{i-1}^{p-1}(t) & \forall i \in \{1, ..., p-1\} \\ tB_{i-1}^{p-1}(t) & i = p \end{cases} \tag{1.2}$$

Proof. We suppose $p \geq 1$ and $i \in \{0, ..., p\}$.

We remind two binomial coefficients properties: $C_p^i = C_{p-1}^i + C_{p-1}^{i-1}$ and $C_p^0 = C_p^p = 1$.

- if i=0:

$$\begin{aligned} B_0^p(t) &= C_p^0 t^0 (1-t)^p \\ &= (1-t)^p \\ &= (1-t)(1-t)^{p-1} \\ &= (1-t)B_0^{p-1}(t) \end{aligned}$$

- if i=p:

$$\begin{aligned} B_p^p(t) &= C_p^p t^p (1-t)^0 \\ &= t^p \\ &= tt^{p-1} \\ &= tB_{p-1}^{p-1}(t) \end{aligned}$$

- <u>if $i \in \{1, ..., p-1\}$</u>

$$\begin{aligned}
B_i^p(t) &= C_p^i t^i (1-t)^{p-i} \\
&= C_{p-1}^i t^i (1-t)^{p-i} + C_{p-1}^{i-1} t^i (1-t)^{p-i} \\
&= (1-t) \underbrace{C_{p-1}^i t^i (1-t)^{p-1-i}}_{=B_i^{p-1}(t)} + t C_{p-1}^{i-1} t^{i-1} (1-t)^{p-i} \\
&= (1-t) B_i^{p-1}(t) + t \underbrace{C_{p-1}^{i-1} t^{i-1} (1-t)^{p-1-(i+1)}}_{=B_{i-1}^{p-1}(t)} \\
&= (1-t) B_i^{p-1}(t) + t B_{i-1}^{p-1}(t)
\end{aligned}$$

□

1.2 Bézier curves

A Bézier curve is a parametric curve defined by these **control points** $P_0, P_1, ..., P_p$, where k is Bézier curve degree.

The mathematical basis for Bézier curves are the Berstein basis polynomial which give us an explicit parmametric equation of the curve $\mathcal{C}$:

$$C(t) = \sum_{i=0}^{p} P_i \cdot B_i^p(t) \tag{1.3}$$

with $B_i^p(t), \forall i \in \{0, ..., p\}$ the p+1 Berstein basis polynomials. So we can say that a Bézier curve has a bernstein polynomial form.

Property 1.2.1.
1. We have $C(0) = P_0$ and $C(1) = P_p$
2. The line $P_0 P_1$ (or $P_{p-1} P_p$) is tangent to $\mathcal{C}$ in P_0 (or P_p).
3. $\mathcal{C}$ is in the convex envelope of $(P_i)_{i \in \{0, ..., p\}}$

Proof.
1. We know that $\forall t \in [0, 1]$, $B_0^0(t) = t^0 (1-t)^0 = 1$.
We define by the recurence formula (1.2) $B_i^p(0)$ and $B_i^p(1)$:

$$\forall p \geq 1 \quad B_i^p(0) = \begin{cases} 0 & i = p \\ B_i^{p-1} & i \in \{0, ..., p-1\} \end{cases}$$

$$\forall p \geq 1 \quad B_i^p(1) = \begin{cases} 0 & i = 0 \\ B_{i-1}^{p-1} & i \in \{1, ..., p\} \end{cases}$$

So we have:

$$\begin{aligned} C(0) &= \sum_{i=0}^{p} P_i B_i^p(0) \\ &= \sum_{i=0}^{p-1} P_i B_i^p(0) + \underbrace{P_p B_p^p(0)}_{=0} \\ &= \dots \\ &= P_0 B_0^1(0) + \underbrace{P_1 B_1^1(0)}_{=0} \\ &= P_0 \underbrace{B_0^0(0)}_{=1} \\ &= P_0 \end{aligned}$$

In the same way, we have:

$$\begin{aligned} C(1) &= \sum_{i=0}^{p} P_i B_i^p(1) \\ &= \sum_{i=1}^{p} P_i B_i^p(1) + \underbrace{P_0 B_0^p(1)}_{=0} \\ &= \sum_{i=1}^{p} P_i B_{i-1}^{p-1}(1) \\ &= \sum_{i=0}^{p-1} P_{i+1} B_i^{p-1}(1) \\ &= \dots \\ &= P_p \underbrace{B_0^0(1)}_{=1} \\ &= P_p \end{aligned}$$

□

1.3 Application : A quadratic Bezier curve

It is easy for the three first order to define **Bezier curves** by calculating corresponding Berstein basis polynomials.

We choose three control points: $P_0 = (0,0)$, $P_1 = (0.3,0)$ and $P_2 = (-0.1, 0.1)$. We will generate the quadratic Bezier curve associated to these points.

We can find Bernstein basis polynomials for order p = 2 by using (1.1) formula:

$$\forall t \in [0,1] \begin{cases} B_0^2(t) = (1-t)^2 \\ B_1^2(t) = 2t(1-t) \\ B_2^2(t) = t^2 \end{cases}$$

We can deduce a parametric equation of our curve by using (1.3):

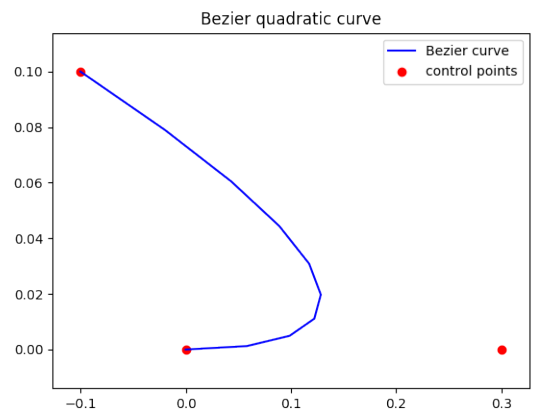

FIGURE 1.1: A Bézier quadratic curve $\mathcal{C}$

$$\begin{aligned} \mathcal{C}(t) &= \sum_{i=0}^{k} P_i \cdot B_i^k \\ &= \sum_{i=0}^{2} P_i \cdot B_i^k \\ &= P_0(1-t)^2 + 2P_1 t(1-t) + P_2 t^2 \end{aligned}$$

Chapter 2

An introduction about B-Splines

2.1 Definition and properties of B-Splines

We start by giving some definitions and properties of B-Splines that we will use later in this report.

2.1.1 First definition of B-Splines

At first, we define $t_0 \leq t_1 \leq ... \leq t_{m-1}$ an increasing sequence of reals. According to Basdevant, 2012 we can define a **B-Splines family** of degree p (note as N_i^p) by recurence:

$$\forall 0 \leq i \leq m-1 \quad \begin{cases} N_i^0(t) = 1 & t \in [t_i, t_{i+1}[\\ N_i^0(t) = 0 & if\ not \end{cases} \tag{2.1}$$

And

$$\forall p \geq 1\ and\ i \leq m-p-1 \quad N_i^p(t) = \frac{t-t_i}{t_{i+p}-t_i} N_i^{p-1}(t) + \frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}} N_{i+1}^{p-1}(t)$$

We choose to note $w_i^p = \frac{t-t_i}{t_{i+p}-t_i}$ to simply our formula. Besides, we have:

$$\begin{aligned} \frac{t_{i+p+1}-t}{t_{i+p+1}-t_{i+1}} &= \frac{t_{i+p+1}-t+t_{i+1}-t_{i+1}}{t_{i+p+1}-t_{i+1}} \\ &= 1 - \frac{t-t_{i+1}}{t_{i+p+1}-t_{i+1}} \\ &= 1 - w_{i+1}^p \end{aligned}$$

So, we can finlly consider

$$\forall p \geq 1\ and\ i \leq m-p-1 \quad N_i^p(t) = w_i^p N_i^{p-1}(t) + (1-w_{i+1}^p) N_{i+1}^{p-1}(t) \tag{2.2}$$

Hypothesis: *We will use the next mathematical convention during these calculation:* $\frac{x}{0} = 0$

Property 2.1.1. 1. $N_i^p(t) = 0$ if $t \notin [t_i, t_{i+p+1}[$ because N_i^p is of compact support

2. partition of unity:

- $\sum_{i=j-p}^{j} N_i^p(t) = 1 \quad t \in [t_j, t_{j+1}[$

- $\sum_{i=0}^{m-p-1} N_i^p(t) = 1 \quad t_p \leq t < t_{m-p}$
- $\sum_{i=0}^{n-1} N_i^p(t) = 1 \quad n \leq m-p \quad and\ t_p \leq t < t_n$

Proof. 1. **compact support:**
Keeping the same notations as above, we choose $\forall p \geq 1$, $0 \leq i \leq m-p-1$, $\forall t \in [0,1]$

$$\begin{aligned} N_i^p(t) &= w_i^p N_i^{p-1}(t) + (1-w_{i+1}^p) N_{i+1}^{p-1}(t) \\ &= w_i^p (w_i^{p-1} N_i^{p-2}(t) + (1-w_{i+1}^{p-1}) N_{i+1}^{p-2}(t)) \\ &+ (1-w_{i+1}^p)(w_{i+1}^{p-1} N_{i+1}^{p-2}(t) + (1-w_{i+2}^{p-1}) N_{i+2} p - 2(t)) \\ &= w_i^p w_i^{p-1} N_i^{p-2}(t) + (w_i^p (1-w_{i+1}^{p-1}) + w_{i+1}^{p-1}(1-w_{i+1}^p) N_{i+1}^{p-2}(t) \\ &+ (1-w_{i+1}^p)(1-w_{i+2}^{p-1}) N_{i+2}^{p-2}(t) \end{aligned} \tag{2.2}$$

We see by recurence $N_i^p(t)$ is a linear combination of $(N_j^0(t))_{j \in \{i,...,i+p\}}$ and we note α_m, $m \in \{i, ..., i+p\}$ the non-zero coefficients such that:

$$N_i^p(t) = \sum_{m=i}^{i+p} \alpha_m N_m^0(t)$$

and such that $N_m^0(t) = \mathbf{1}_{[t_m, t_{m+1}[}(t)$

So we have : $supp(N_i^p) = \cup_{m=i}^{i+p} [t_m, t_{m+1}[= [t_i, t_{i+p+1}[$

2. **partition unity:**

- $\underline{\sum_{i=j-p}^{j} N_i^p(t) = 1 \quad t \in [t_j, t_{j+1}[:}$

$$\begin{aligned} \sum_{i=j-p}^{j} N_i^p(t) &= N_{j-p}^p(t) + N_{j-p+1}^k(t) + ... + N_{j-1}^p(t) + N_i^p(t) \\ &= (w_{j-p}^p N_{j-p}^{p-1} + (1-w_{j-p+1}^p) N_{j-p+1}^{p-1}(t)) + (w_{j-p+1}^p N_{j-p+1}^{p-1} + (1-w_{j-p+2}^p) N_{j-p+2}^{p-1} \\ &+ (w_{j-1}^p N_{j-1}^{p-1} + (1-w_j^p) N_j^{p-1}(t)) + (w_j^p N_j^{p-1} + (1-w_{j+1}^p) N_{j+1}^{p-1}(t)) \\ &= w_{j-p}^p \underbrace{N_{j-p}^{p-1}(t)}_{=0} + \sum_{i=j-p+1}^{j} N_i^{p-1}(t) + \underbrace{(1-w_{j+1}^p) N_{j+1}^{p-1}(t)}_{=0} \end{aligned}$$

because $supp(N_{j+1}^{p-1}) = [t_{j+1}; t_{j+p+1}[$ and $[t_{j+1}; t_{j+p+1}[\cap [t_j; t_{j+1}[= \oslash$ just as N_{j-p}^{p-1}

$$\sum_{i=j-p}^{j} N_i^p(t) = \sum_{i=j-p+1}^{j} N_i^{p-1}(t)$$

and by recurence

$$\sum_{i=j-p}^{j} N_i^p(t) = N_j^0(t) = 1$$

- $\underline{\sum_{i=0}^{m-p-1} N_i^p(t) = 1 \quad t_p \leq t < t_{m-p}:}$

$$\begin{aligned}
\sum_{i=0}^{m-p-1} N_i^p(t) &= \sum_{i=0}^{m-p-1} w_i^p N_i^{p-1}(t) + (1 - w_{i+1}^{p-1}(t) \\
&= \sum_{i=0}^{m-p-1} w_i^p N_i^{p-1}(t) + \sum_{i=0}^{m-p-1} N_{i+1}^{p-1}(t) - \sum_{i=0}^{m-p-1} w_{i+1}^p N_{i+1}^{p-1}(t) \\
&= w_0^p \underbrace{N_0^{p-1}(t)}_{=0} + w_{m-p}^p \underbrace{N_{m-p}^{p-1}(t)}_{=0} + \sum_{i=0}^{m-p-1} N_{i+1}^{p-1}(t) \\
&= \sum_{i=0}^{m-p-1} N_{i+1}^{p-1}(t) \\
&= ... \\
&= N_p^0(t) \\
&= 1
\end{aligned}$$

- $\underline{\sum_{i=0}^{n-1} N_i^p(t) = 1 \quad n \leq m-p \quad \textit{and } t_p \leq t < t_n}$
  same as above

□

Property 2.1.2. we can define the derivative of the function $N_i^p(t)$ by:

$$(N_i^p)'(t) = \frac{p}{t_{i+p} - t_i} N_i^{p-1}(t) - \frac{p}{t_{i+1+p} - t_{i+1}} N_{i+1}^{p-1}(t)$$

and we get T* the knotlist associated to $(N_i^p)'(t)_i$ and we have T* defined from the N_i^p knotlist T: T* = $[t_1, ..., t_{m-2}]$. Besides as we derivate $N_i^p(t)$ which is a polynomial function of degree p, then $(N_i^p)'(t)$ is a polynomial function of degree p-1.

This property will be very often used in the finite element method, especially when we will calculate the mass matrix

Property 2.1.3. We consider T a knots list of m knots and we choose to create B-Splines of degree p. So our B-Splines family will be m - p - 1 members.

2.1.2 B-Splines curves

They are a generalization of Bézier curves.

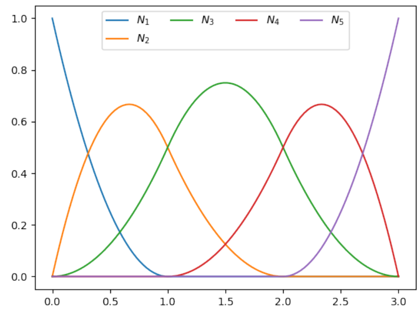

FIGURE 2.1: Representation of a order 2 B-Splines family with T = [0,0,0, 1, 2, 3,3, 3]

A **Spline curve of degree p** is a parametric curve of B-Spline functions of degree p based on some control points $P_0, P_1, ..., P_{n-1}$.

$$\begin{array}{l} S : [0,1] \to \mathbb{R}^d \\ S(t) = \sum_{i=0}^{n-1} N_i^p(t) \cdot P_i \quad t \in [0,1] \end{array} \tag{2.3}$$

We present here a basis example of a B-Splines curve based on some control points

We will create a B-Splines curve based 9 knots : $T = [0,0,1,2,3,4,5,6,6]$ and 7 control points $P_0, ..., P_6$ so we are looking for a quadratic B-Spline curve because we know that:

$$\begin{aligned} p &= m - n \\ &= 9 - 7 \\ &= 2 \end{aligned}$$

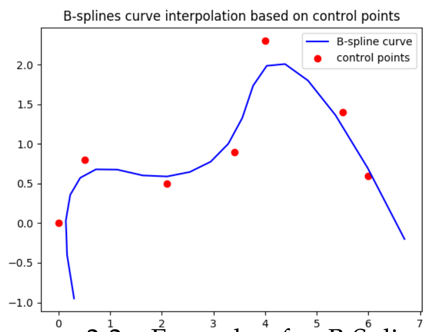

FIGURE 2.2: Example of a B-Spline Curve based on a knot list and control points

$$\begin{aligned} P_0 &= (0,0) & P_4 &= (4,2.3) \\ P_1 &= (0.5,0.8) & P_5 &= (5.5,1.4) \\ P_2 &= (2.1,0.5) & P_6 &= (6,0.6) \\ P_3 &= (3.4,0.9) \end{aligned}$$

2.2 Resolution of a Poisson 1D equation

We should start by reminding Poisson equation in 1D:

$$\begin{cases} -\Delta u = f \quad in\ \Omega = [0,1] \\ u = 0 \quad on\ \partial\Omega \end{cases} \tag{2.4}$$

As we want to use a **finite elements method**, we first find the variationnal formulation.

We consider $v \in \mathcal{V}_h$ a test function with $\mathcal{V}_h$ a test space not defined yet

$$\begin{aligned}(2.4) &\iff -\Delta u = f \\ &\iff -\Delta u \cdot v = f \cdot v \\ &\iff -\int_\Omega \Delta u \cdot v = \int_\Omega f \cdot v\end{aligned}$$

By using **Green formula**, we have,

$$(2.4) \iff \int_\Omega \vec{\nabla} u \cdot \vec{\nabla} v - \underbrace{\int_{\partial\Omega} u \cdot v \cdot n}_{=0} = \int_\Omega f \cdot v$$

Finally, we find this variational formulation:

$$\int_\Omega \vec{\nabla} u \cdot \vec{\nabla} v = \int_\Omega f \cdot v, \quad \forall v \in \mathcal{V}_h \tag{2.5}$$

We choose to use B-Splines Family named $(N_i^p)_i$ as $\mathcal{V}_h$ space. By this way, we can consider:

$$\begin{cases} u = \sum_j u_j N_j^p \\ v = N_i^p \end{cases}$$

So we can discretize (2.5) in

$$\sum_j u_j \underbrace{\int_\Omega \vec{\nabla} N_i^p \cdot \vec{\nabla} N_j^p}_{S_{ij}} = \underbrace{\int_\Omega f \cdot N_i^p}_{F_i}$$

With S the stiffness matrix and F a vector which represent the second member discretized.

We have :

$$S = \begin{pmatrix} \int_\Omega \vec{\nabla} N_0^p \cdot \vec{\nabla} N_0^p & \dots & \dots & \int_\Omega \vec{\nabla} N_0^p \cdot \vec{\nabla} N_{m-1}^p \\ \int_\Omega \vec{\nabla} N_1^p \cdot \vec{\nabla} N_0^p & \dots & \dots & \int_\Omega \vec{\nabla} N_1^p \cdot \vec{\nabla} N_{m-1}^p \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \int_\Omega \vec{\nabla} N_{m-1}^p \cdot \vec{\nabla} N_0^p & \dots & \dots & \int_\Omega \vec{\nabla} N_{m-1}^p \cdot \vec{\nabla} N_{m-1}^p \end{pmatrix} \in \mathcal{M}_{m\times m}(\mathbb{R})$$

and

$$F = \begin{pmatrix} \int_\Omega f N_0^p \\ \int_\Omega f N_1^p \\ \dots \\ \dots \\ \int_\Omega f N_{m-1}^p \end{pmatrix} \in \mathbb{R}^m$$

We want to solve this linear problem: $Su = F$, with $u \in \mathbb{R}^m$. As we have homogenous boundary conditions, we can change our system by:

$$S' = \begin{pmatrix} \int_\Omega \overrightarrow{\nabla} N_1^p \cdot \overrightarrow{\nabla} N_1^p & \dots & \dots & \int_\Omega \overrightarrow{\nabla} N_1^p \cdot \overrightarrow{\nabla} N_{m-2}^p \\ \int_\Omega \overrightarrow{\nabla} N_2^p \cdot \overrightarrow{\nabla} N_1^p & \dots & \dots & \int_\Omega \overrightarrow{\nabla} N_2^p \cdot \overrightarrow{\nabla} N_{m-2}^p \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ \int_\Omega \overrightarrow{\nabla} N_{m-2}^p \cdot \overrightarrow{\nabla} N_1^p & \dots & \dots & \int_\Omega \overrightarrow{\nabla} N_{m-2}^p \cdot \overrightarrow{\nabla} N_{m-2}^p \end{pmatrix} \in \mathcal{M}_{m-2}(\mathbb{R})$$

and

$$F' = \begin{pmatrix} \int_\Omega f N_1^p \\ \int_\Omega f N_2^p \\ \dots \\ \dots \\ \int_\Omega f N_{m-2}^p \end{pmatrix} \in \mathbb{R}^{m-2}$$

and finally, we want to solve this new problem:

$$S' \cdot u = F' \quad u \in \mathbb{R}^{m-2} \tag{2.6}$$

Numerical results

In this first example, we try to approximate the solution of this system:

$$\begin{cases} -\Delta u = 2 \\ \Omega = [0,1] \\ u(0) = 0 \quad u(1) = 0 \end{cases} \tag{2.7}$$

and we know the analytic solution of this problem which is $u(x) = x(1-x)$

The numerical solution approaches the analytical solution but as we have a long time of computing we use few knots. This is the reason that explains why we will use the **spl module for Python** in the next solvers.

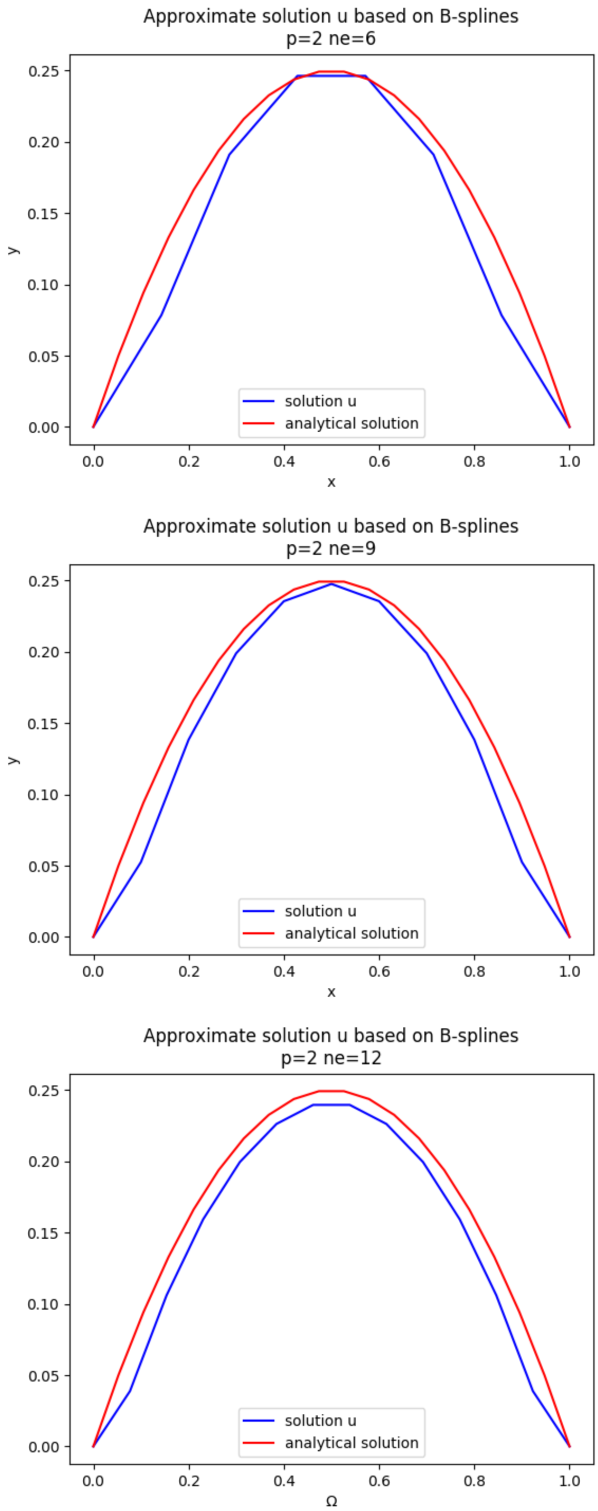

FIGURE 2.3: Approximate solution u for Poisson 1D equation with quadratic B-Splines

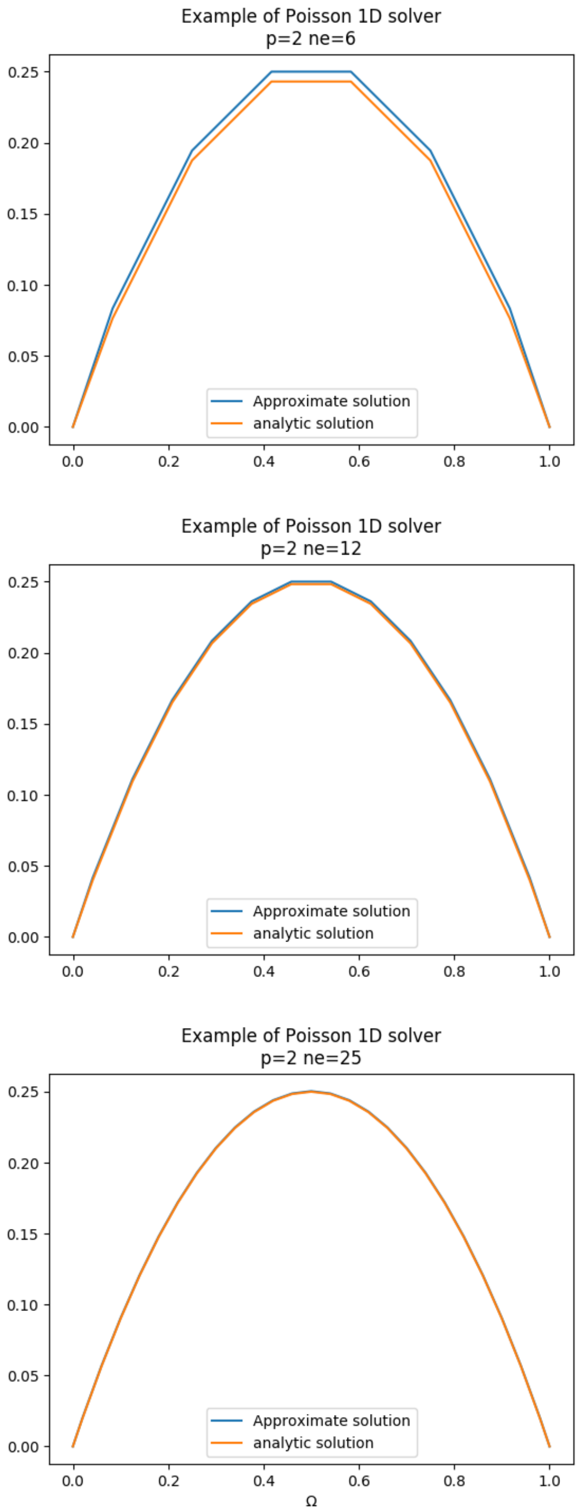

FIGURE 2.4: Approximate solution u for Poisson 1D equation with quadratic B-Splines and SPL module

Chapter 3

Solver of Maxwell's equations 1D

To learn about the equations of Maxwell and about his developments, we had worked on the book of Bossavit, 1998

3.1 Definitions and notations of Maxwell' equations

These next equations are named Maxwell equations:

$$-\partial_t \mathcal{D} + \nabla \times H = j \tag{3.1}$$
$$\partial_t \mathcal{B} + \nabla \times E = 0 \tag{3.2}$$
$$\mathcal{D} = \epsilon_0 E + p \tag{3.3}$$
$$\mathcal{B} = \mu_0 (H + m) \tag{3.4}$$

where the quantities have following meanings:

- E(x, t) is the electric field intensity
- $\mathcal{B}(x, t)$ is the magnetic flux density
- $\mathcal{D}(x, t)$ is the magnetic induction
- H(x, t) is the magnetic field
- ρ is the charge density
- j(x, t) vector current density
- μ_0 the magnetic permeability
- ϵ_0 the electric permittivity
- $x \in \mathbb{R}^d$ and represent space
- $t \in \mathbb{R}^+$ and represent time

But we can rewrite some of these equations by eliminating $\mathcal{D}$ and H:

$$\begin{aligned} & -\partial_t \mathcal{D} + \nabla \times H = j \\ \Longleftrightarrow & -\partial_t (\epsilon_0 E + \rho) + \nabla \times (\frac{1}{\mu_0} B - m) = j \\ \Longleftrightarrow & -\epsilon_0 \partial_t E + \frac{1}{\mu_0} \nabla \times B = J - \partial_t \rho + \nabla \times m \end{aligned}$$

and if we simplify our equations with $p = 0$ and $m = 0$, so we get:

$$\begin{cases} -\epsilon_0 \partial_t E + \frac{1}{\mu_0} \nabla \times B = j \\ \partial_t \mathcal{B} + \nabla \times E = 0 \\ \mathcal{D} = \epsilon_0 E \\ \mathcal{B} = \mu_0 H \end{cases}$$

As we have an explicit expression for $\mathcal{D}$ and H, we can study more precisely the two other equations which become a system.

$$\begin{cases} -\epsilon_0 \partial_t E + \frac{1}{\mu_0} \nabla \times B = j \\ \partial_t \mathcal{B} + \nabla \times E = 0 \end{cases} \tag{3.5}$$

We can also add some other laws to help us in our calculation:

$$\begin{cases} \nabla \times E(x,t) = -\partial_t \mathcal{B}(x,t) & (Faraday's\ law) \quad (3.6) \\ \nabla \cdot E(x,t) = \dfrac{\rho(x,t)}{\epsilon_0} & (Gauss'\ law) \quad (3.7) \\ \nabla \times \mathcal{B}(x,t) = \mu_0 j(x,t) + \mu_0 \epsilon_0 \partial_t E(x,t) & (Ampere's\ law) \quad (3.8) \\ \nabla \cdot \mathcal{B}(x,t) = 0 & (Gauss'\ law\ for\ magnetism) \quad (3.9) \end{cases}$$

Our aim will be to find an approximation of E and $\mathcal{B}$ and we consider j, m and ρ as term-sources of our system.

3.2 Reducing Maxwell's equations to a 1D problem

A particular attention is paid to the Faraday's law and the Ampere's law:

$$(3.6) \iff \begin{cases} \partial_y E_z - \partial_z E_y + \partial_t \mathcal{B}_x = 0 \\ \partial_z E_x - \partial_x E_z + \partial_t \mathcal{B}_y = 0 \\ \partial_x E_y - \partial_y E_x + \partial_t \mathcal{B}_z = 0 \end{cases}$$

and

$$(3.8) \Leftrightarrow \begin{cases} -\epsilon_0 \partial_t E_x + \frac{1}{\mu_0}(\partial_y \mathcal{B}_z - \partial_z \mathcal{B}_y) = j_x \\ -\epsilon_0 \partial_t E_y + \frac{1}{\mu_0}(\partial_z \mathcal{B}_x - \partial_x \mathcal{B}_z) = j_y \\ -\epsilon_0 \partial_t E_z + \frac{1}{\mu_0}(\partial_x \mathcal{B}_y - \partial_y \mathcal{B}_x) = j_z \end{cases}$$

For a x-directed, y-polarized electromagnetic wave, (so $E_x = \mathcal{B}_x = 0$), there is no variation in the x, y direction so $\frac{\partial}{\partial y} = \frac{\partial}{\partial z} = 0$. We can reduce the two previous system to :

$$\begin{cases} \partial_x E_y + \partial_t \mathcal{B}_z = 0 \\ \epsilon_0 \partial_t E_y + \frac{1}{\mu_0} \partial_x \mathcal{B}_z = -j_y \end{cases} \tag{3.10}$$

For the purposes of notation, we consider E_y as E, $\mathcal{B}_z$ as $\mathcal{B}$ and j_y as j

3.3 Discretization of the electric/magnetic field system

In this part we would like to solve Maxwell equations $\forall x \in \Omega \subseteq \mathbb{R}$. By reminding our minimal system (3.10), and as we have $x \in \mathbb{R}$, it becomes:

$$\begin{cases} \epsilon_0 \partial_t E(x,t) + \dfrac{1}{\mu_0} \partial_x \mathcal{B}(x,t) = -j & (3.11) \\ \partial_t \mathcal{B}(x,t) + \partial_x E(x,t) = 0 & (3.12) \end{cases}$$

Firstly, we take a look on (3.11), and we choose v a test function in $\mathbb{S}^p$ space [1]

$$\begin{aligned} (3.11) &\iff \int_\Omega \epsilon_0 \partial_t E \cdot v + \frac{1}{\mu_0} \partial_x \mathcal{B} \cdot v dx = \int_\Omega -j \cdot v dx \\ &\iff \int_\Omega \epsilon_0 \partial_t E \cdot v dx + \frac{1}{\mu_0} \int_\Omega \partial_x \mathcal{B} \cdot v dx = \int_\Omega -j \cdot v dx \\ &\iff \int_\Omega \epsilon_0 \partial_t E \cdot v dx + \frac{1}{\mu_0} \left(\underbrace{[\mathcal{B} \cdot v]}_{=0} - \int_\Omega \mathcal{B} \cdot \partial_x v \right) dx = \int_\Omega -j \cdot v dx \end{aligned}$$

and so

$$(3.11) \iff \int_\Omega \epsilon_0 \partial_t E \cdot v dx - \frac{1}{\mu_0} \int_\Omega \mathcal{B} \cdot \partial_x v dx = \int_\Omega -j \cdot v dx$$

We start by discretizing Ω in some breakpoints $x_0, x_1, ..., x_r$. Based on these we define an **uniform clamped list of knots** which will help us to create a family of B-Splines of degree p in the Schoenberg space. So we build our list of non-decreasing knots $\chi_0, ..., \chi_{m-1}$ and $\chi_0 = \chi_1 = ... = \chi_p = x_0$ just as $\chi_{m-1-p} = \chi_{m-p} = ... = \chi_{m-1} = x_r$

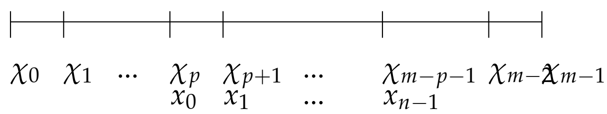

FIGURE 3.1: Ω discretization in 1D case

We have created our family of B-Splines with m knots so we will have $n = m - p$ B-Splines.

We can know discretize the equation 3.11 by using a family of B-Splines $(N_i^p)_{0 \le i \le n-1}$ of $\mathbb{S}^p$ and $(D_i^{p-1})_{0 \le i \le n-1}$ of $\mathbb{S}^{p-1}$:

- $\partial_t E(x,t) = \sum_{j=0}^{n-1} \partial_t E(\overline{x}_j, t) N_j^p(x) = \sum_{j=0}^{n-1} \dot{E}_j N_j^p(x)$
- $v(x) = N_i^p(x)$
- $\mathcal{B}(x) = \sum_{j=0}^{n-2} \mathcal{B}(\overline{x}_j, t) D_j^{p-1}(x) = \sum_{j=0}^{n-2} \mathcal{B}_j D_j^{p-1}(x)$

[1] the Schoenberg space of degree p. cf: Appendix D for more details

where $\overline{x}_j$ are **the Greville abscissae**[2] of our problem.

with $D_i^{p-1}(x) \in \mathbb{S}^{p-1}$, $\forall i \in \{0, ..., n-1\}$ an other family of B-Spline, because of the size of $\mathcal{B} \in \mathcal{R}^{n-1}$. So:

$$(3.11) \iff$$

$$\sum_{j=0}^{n-1} \epsilon_0 \dot{E}_j \int_\Omega N_j^p(x) N_i^p(x) dx - \frac{1}{\mu_0} \sum_{j=0}^{n-2} \mathcal{B}_j \int_\Omega D_j^{p-1}(x)(N_i^p(x))' dx = \sum \int_\Omega -j(\overline{x}_i) \cdot N_i^p(x) dx$$

Then we find this system:

$$\epsilon_0 M \dot{E} - \frac{1}{\mu_0} RB = J \tag{3.13}$$

with

$$M \in \mathcal{M}_n(\mathbb{R}),\ M_{ij} = \int_\Omega N_i^k(x) N_j^k(x) dx$$

which the **mass matrix of the space** $\mathbb{S}^p$

$$R \in \mathcal{M}_{n,n-1}(\mathbb{R}),\ R_{ij} = \int_\Omega D_j^{k-1}(x)(N_i^k(x))' dx$$

$$J \in \mathbb{R}^p,\ J_i = \int_\Omega -j(\overline{x}_i) \cdot N_i^p(x) dx$$

and $\dot{E} \in \mathbb{R}^n, \dot{E}_i = \partial_t E(\overline{x}_i, t)$ and $\mathcal{B} \in \mathbb{R}^{n-1}, \mathcal{B}_i = \mathcal{B}(\overline{x}_i, t)$.

Now, we can discretize our second equation (3.12) by using the same method:

- $\partial_t \mathcal{B}(x,t) = \sum_{j=0}^{n-2} \dot{\mathcal{B}}_j D_j^{p-1}(x)$
- $\partial_x E(x,t) = \sum_{j=0}^{n-1} E_j (N_j^p)'(x)$

and so:

$$(3.12) \iff \sum_{j=0}^{n-2} \dot{\mathcal{B}}_j D_j^{p-1}(x) + \sum_{j=0}^{n-1} E_j (N_j^p)'(x) = 0$$

Knowing that $D_i^{p-1} \in \mathbb{S}^{p-1}$ we want to express it as a linear combination of $(N_i^{p-1})_i$ family which is the B-Splines family of reference in $\mathbb{S}^{p-1}$

According to the property of derivative B-splines we know:

$$(N_i^p)'(x) = \frac{p}{t_{i+p} - t_i} N_i^{p-1}(x) - \frac{p}{t_{i+p+1} - t_{i+1}} N_{i+1}^{p-1}(x)$$

$$\begin{aligned}(3.12) &\iff \sum_{j=0}^{n-2} \dot{\mathcal{B}}_j D_j^{p-1}(x) + \sum_{j=0}^{n-1} E_j (N_j^p)'(x) = 0 \\ &\iff \sum_{j=0}^{n-2} \dot{\mathcal{B}}_j D_j^{p-1}(x) + \sum_{j=0}^{n-1} E_j \left(\frac{p}{t_{i+p} - t_i} N_i^{p-1}(x) - \frac{p}{t_{i+p+1} - t_{i+1}} N_{i+1}^{p-1}(x) \right) = 0\end{aligned}$$

[2] cf. Appendix E for more details

If we choose $D_i^{p-1} = \frac{p}{t_{i+p}-t_i} N_i^{p-1}(x)$ then we get:

$$\Longleftrightarrow \sum_{j=0}^{n-2} \dot{\mathcal{B}}_j D_j^{p-1}(x) + \sum_{j=0}^{n-1} E_j \left(D_j^{p-1}(x) - D_{j+1}^{p-1}(x) \right) = 0$$

and we can transform this:

$$\begin{aligned}
\sum_{j=0}^{n-1} E_j \left(D_j^{p-1}(x) - D_{j+1}^{p-1}(x) \right) &= \sum_{j=0}^{n-1} E_j D_j^{p-1}(x) - \sum_{j=0}^{n-1} E_j D_{j+1}^{p-1}(x) \\
&= \sum_{j=0}^{n-1} E_j D_j^{p-1}(x) - \sum_{j=1}^{n} E_{j-1} D_j^{p-1}(x) \\
&= E_0 D_0^{p-1}(x) + \sum_{j=1}^{n-1} (E_j - E_{j-1}) D_j^{p-1}(x) + E_{n-1} D_n^{p-1}(x)
\end{aligned}$$

But we know $D_0^{p-1}(x) = \frac{p}{k_p - k_0} N_0^{p-1}(x)$ and $k_p = k_0$ so according to our hypothesis, we have $\frac{p}{k_p-k_0} = \frac{p}{0} = 0$ and finally $D_0^{p-1}(x) = 0$

In the same way, $D_n^{p-1}(x) = \frac{p}{k_n - k_{n+p}} N_n^{p-1} = 0$

$$\begin{aligned}
\sum_{j=0}^{n-1} E_j \left(D_j^{p-1}(x) - D_{j+1}^{p-1}(x) \right) &= E_0 D_0^{p-1}(x) + \sum_{j=1}^{n-1} (E_j - E_{j-1}) D_j^{p-1}(x) + E_{n-1} D_n^{p-1}(x) \\
&= \sum_{j=1}^{n-1} (E_j - E_{j-1}) D_j^{p-1}(x)
\end{aligned}$$

So

$$(3.12) \Longleftrightarrow \sum_{j=0}^{n-2} \dot{\mathcal{B}}_j D_j^{p-1}(x) + \sum_{j=1}^{n-1} (E_j - E_{j-1}) D_j^{p-1} = 0$$

We can transform this in a system:

$$\dot{\mathcal{B}} - GE = 0 \tag{3.14}$$

where $G = \begin{pmatrix} 1 & -1 & 0 & ... & 0 & 0 \\ 0 & 1 & -1 & ... & 0 & 0 \\ ... & ... & ... & ... & ... & 0 \\ 0 & 0 & 0 & ... & 1 & -1 \end{pmatrix} \in \mathcal{M}_{n-1,n}(\mathbb{R})$

and $\dot{\mathcal{B}} \in \mathbb{R}^{n-1}$.

Finally, our system becomes:

$$\begin{cases} \epsilon_0 M\dot{E} - \frac{1}{\mu_0} RB = J \quad (3.13) \\ \dot{B} - GE = 0 \quad (3.14) \end{cases} \tag{3.15}$$

Assembly of the matrices

We start by assembling the mass matrix of the B-Spline space $\{N_0^p, N_1^p, ..., N_{m-p-1}^p$.

We consider the discretization of $\Omega = [\chi_0;\chi_1[\cup[\chi_1;\chi_2[\cup ... \cup [\chi_{m-2};\chi_{m-1}[$ and we name $e_0 = [\chi_0;\chi_1[$, $e_i = [\chi_i;\chi_{i+1}[$ with $i \in \{0, ..., m-2\}$

We have :

$$\begin{aligned} M_{ij} &= \int_\Omega N_i^p(x) \cdot N_j^p(x)dx \\ &= \sum_{l=0}^{m-2} \int_{e_l} N_i^p(x) \cdot N_j^p(x)dx \end{aligned}$$

Besides, we know that the support of N_i^p is $[\chi_i;\chi_{i+p+1}[= e_i \cup e_{i+1} \cup ... \cup e_{i+p}$, so we deduce that there are p+1 non-vanishing B-Splines on a element e.

In the same way we can build an algorithm for the rotational matrix R.

To verify those matrices we look for a relation between R and G:

$$\begin{aligned} R_{ij} &= \int_\Omega (N_i^p)' D_j^{p-1}(x)dx \\ &= \int_\Omega (D_i^{p-1}(x) - D_{i+1}^{p-1}) \cdot D_j^{p-1}(x)dx \\ &= \int_\Omega D_i^{p-1}(x) \cdot D_j^{p-1}(x)dx - \int_\Omega D_{i+1}^{p-1} \cdot D_j^{p-1}(x)dx \end{aligned}$$

We note $M_\mathcal{B}$ the mass matrix of the space $W = span(D_i^{p-1}, \forall i \in \{0, ..., n-1\})$ and as $G_{ii} = 1$ and $G_{i,i-1} = -1, \forall i \geq 1$, so

$$\begin{aligned} R_{ij} &= \int_\Omega D_j^{p-1}(x) \cdot D_i^{p-1}(x)dx \cdot G_{ii} + \int_\Omega D_j^{p-1}(x) \cdot D_{i+1}^{p-1}(x)dx \cdot G_{i+1,i} \\ &= (M_\mathcal{B})_{ij} G_{ii} + (M_\mathcal{B})_{j,i+1} G_{i+1,i} \\ &= \sum_{k=0}^{m-p-2} (M_\mathcal{B})_{jk} G_{ki} \\ &= (M_\mathcal{B} G)_{ji} \\ \iff M_\mathcal{B} G &= R^T \end{aligned}$$

Algorithm 1 Compute the mass matrix of the V space by using SPL package for Python

Require: V spline space of degree p
Ensure: $M \in \mathcal{M}_{n,n}(\mathbb{R})$, $M_{ij} = \int_\Omega N_i^k(x) \cdot N_j^k(x) dx$

```
M ← StencilMatrix(V)
for element in liste of elements do
  for i_spline in element do
    for j_spline in element do
      res ← 0.0
      for each quadrature point in element do
        nik ← V.quad_basis[element, i_spline, 0, quadrature point]
        njk ← V.quad_basis[element, j_spline, 0, quadrature point]
        w ← V.quad_weight[element, quadrature point]
        res ← res × (nik × njk × w)
      end for
      M(i, j) ← M(i, j) + res
    end for
  end for
end for
```

Algorithm 2 Compute the rotational matrix of the V space by using SPL package for Python

Require: V spline space of degree p and W spline space of degree p-1
Ensure: $R \in \mathcal{M}_{n,n-1}(\mathbb{R})$, $R_{ij} = \int_\Omega (N_i^k(x))' \cdot D_j^{k-1}(x) dx$

```
n ← len(V.knots) − p − 1
M ← zeros((n, n − 1))
for element in liste of elements do
  for i_spline in element do
    for j_spline in element do
      res ← 0.0
      for each quadrature point in element do
        nik ← V.quad_basis[element, i_spline, 1, quadrature point]
        njk ← W.quad_basis[element, j_spline, 0, quadrature point]
        alpha_j ← p / (W.knots[j_spline+p] − W.knots[j_spline])
        djk ← alpha_j × njk
        w ← V.quad_weight[element, quadrature point]
        res ← res × (nik × djk × w)
      end for
      R(i, j) ← R(i, j) + res
    end for
  end for
end for
```

3.4 Commuting Projectors

We would like to find different commuting projectors able to keep function properties even when we discretize them.

3.4.1 A first approach

In this part we want to define a projector able to discretize a function f on Ω in the N_i^p span. $\pi_{l^2}(f) = \sum_{j=0}^{m-p-1} f_j \cdot N_j^p(x)$

We consider here $v(x) = N_i^p(x)$ a test function for any $i \in \{0, ..., m-p-1\}$

$$\int_\Omega \pi_{L^2}(f) \cdot v dx = \int_\Omega f \cdot v dx$$

So we looking for $(f_j)_j$ as

$$\sum_{j=0}^{m-p-1} f_j \int_\Omega N_j^p(x) \cdot N_i^p(x) dx = \int_\Omega f(x) \cdot N_i(x) dx$$

which mean we want to solve the next system:

$$M \cdot \begin{pmatrix} f_0 \\ f_1 \\ ... \\ f_{m-p-1} \end{pmatrix} = \begin{pmatrix} \int_\Omega f(x) \cdot N_0^p(x) \\ \int_\Omega f(x) \cdot N_1^p(x) \\ ... \int_\Omega f(x) \cdot N_{m-p-1}^p(x) \end{pmatrix} \tag{3.16}$$

where M is the mass matrix presented before.

We find $(f_j)_j$ and we notice that $||f_j - \overline{f_j}|| \longrightarrow 0$ so π_{L^2} can approximate the Greville abscissae of a function

$$\begin{tikzcd} H^1(\Omega) \arrow[r, "\mathbf{d}"] \arrow[d, "\pi_{H^1}"'] & L^2 \arrow[d, "\pi_{L^2}"] \\ V_h(grad, \Omega) \arrow[r, "G"] & V_h(L^2, \Omega) \end{tikzcd}$$

FIGURE 3.2: DeRham sequence 1D

The two next projectors are also presented in the future article Manni et al., unpublished yet. This work is here an example to the theory of this article.

3.4.2 Projector π_{H^1}

We consider this first projector :

$$\begin{cases} u_\pi^0(\chi_i) = u(\chi_i), & \forall \chi \in \Omega, \forall u \in V_h(grad, \Omega) \\ \pi_{H^1} u = u_\pi^0 \end{cases}$$

As we have $u_\pi^0 u \in V$, then $u_\pi^0 u = \sum_{j=0}^{n-1} u_j N_j^p(x)$ and so :

$$
\begin{aligned}
& u_\pi^0(\chi_i) = u(\chi_i) \\
\Longleftrightarrow & \sum_{j=0}^{n-1} u_j N_j^p(\chi_i) = u(\chi_i) \\
\Longleftrightarrow & \begin{pmatrix} N_0^p(\chi_0) & N_1^p(\chi_0) & ... & N_{n-1}^p(\chi_0) \\ ... & ... & ... & ... \\ N_0^p(\chi_{n-1}) & N_1^p(\chi_{n-1}) & ... & N_{n-1}^p(\chi_{n-1}) \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ ... \\ u_{n-1} \end{pmatrix} = \begin{pmatrix} u(\chi_0) \\ ... \\ u(\chi_{n-1}) \end{pmatrix}
\end{aligned}
$$

By solving this system, we can find the vector $(u_0, u_1, ..., u_{n-1})^T$ which is the projection of the function $u \in H^1$ into $V_h(grad, \Omega)$ space

3.4.3 Projetor π_{L^2}

Let's take a look on the π_{L^2} projector which is defined as:

$$
\begin{cases} \pi_{L^2} u(x) = u_\pi^1(x) \in V_h(L^2, \Omega) \\ \int_{\chi_i}^{\chi_{i+1}} u_\pi^1(y)dy = \int_{\chi_i}^{\chi_{i+1}} u(y)dy \quad \chi_i \in \Omega \end{cases}
$$

As we have here $\pi_{L^2} u(x) \in V_h(L^2, \Omega)$ then $\pi_{L^2} u(x) = \sum_{j=0}^{n-2} u_j D_j^{p-1}(x)$, so

$$
\begin{aligned}
& \int_{\chi_i}^{\chi_{i+1}} u_\pi^1(y)dy = \int_{\chi_i}^{\chi_{i+1}} u(y)dy \\
\Longleftrightarrow & \int_{\chi_i}^{\chi_{i+1}} \sum_{j=0}^{n-2} u_j D_j^{p-1}(y)dy = \int_{\chi_i}^{\chi_{i+1}} u(y)dy \\
\Longleftrightarrow & \sum_{j=0}^{n-2} u_j \int_{\chi_i}^{\chi_{i+1}} D_j^{p-1}(y)dy = \int_{\chi_i}^{\chi_{i+1}} u(y)dy \\
\Longleftrightarrow & \begin{pmatrix} \int_{\chi_0}^{\chi_1} D_0^{p-1}(y)dy & \int_{\chi_0}^{\chi_1} D_1^{p-1}(y)dy & ... & \int_{\chi_0}^{\chi_1} D_{n-2}^{p-1}(y)dy \\ ... & ... & ... & ... \\ \int_{\chi_{m-2}}^{\chi_{m-1}} D_0^{p-1}(y)dy & \int_{\chi_{m-2}}^{\chi_{m-1}} D_1^{p-1}(y)dy & ... & \int_{\chi_{m-2}}^{\chi_{m-1}} D_{n-2}^{p-1}(y)dy \end{pmatrix} \cdot \begin{pmatrix} u_0 \\ ... \\ u_{n-2} \end{pmatrix} = \begin{pmatrix} \int_{\chi_0}^{\chi_1} u(y)dy \\ ... \\ \int_{\chi_{m-2}}^{\chi_{m-1}} u(y)dy \end{pmatrix}
\end{aligned}
$$

The solution of this system give us the vector $(u_0, ..., u_{n-2})^T$ which is the projection of the function $u \in L^2$ into $V_h(L^2, \Omega)$ space

3.4.4 Commuting of π_{H^1} and π_{L^2}

We know that for $u \in H^1$ then $d \cdot u = u' \in L^2$ and we would like to prove that their projectors act in the same way,

$$
d \cdot \pi_{H^1} u(x) = \pi_{L^2} u(x) \tag{3.17}
$$

Proof. we consider $u \in H^1(\Omega)$ so we can use π_{H^1} and π_{L^2} projectors on u:

$$\pi_{H^1} u(x) = \sum_{j=0}^{n-1} u_j \cdot N_j^p(x)$$
$$\pi_{L^2} u(x) = \sum_{j=0}^{n-2} u_j \cdot D_j^{p-1}(x)$$

$$\begin{aligned}
d \cdot (\pi_{H^1} u(x)) &= d \cdot (\sum_{j=0}^{n-1} u_j N_j^p(x)) \\
&= \sum_{j=0}^{n-1} u_j d(N_j^p(x)) \\
&= \sum_{j=0}^{n-1} u_j (D_j^{p-1}(x) - D_{j+1}^{p-1}(x)) \\
&= \sum_{j=0}^{n-1} u_j D_j^{p-1}(x) - \sum_{j=0}^{n-1} u_j D_{j+1}^{p-1}(x) \\
&= \sum_{j=0}^{n-1} (u_j - u_{j-1}) D_j^{p-1}(x)
\end{aligned}$$

□

3.5 Some example of scheme

We discretize our system in time and we note $t^0, t^1, ..., t^\tau, ..., t^f$ the differents time and Δt is the time step. So we note $E_i^\tau = E(\overline{x}_i, t^\tau)$

We choose the next example for all these schemes:

- j = 0
- we have homogenous Dirichlet boundary conditions
- $E(x,0) = E^0$ and $B(x,0) = B^0$ are known

Our aim is to find experimentally the electric field E and the magnetic field B whose analytical solutions are :

- $E(x,t) = \sin(2\pi x)\cos(t)$
- $B(x,t) = \cos(2\pi x)\sin(t)$

To discretize $E(x,0)$ and $B(x,0)$ in the $(N_i^p)_i$ span we use the projector π_{L^2}

3.5.1 Euler Explicit Scheme

This is a first order scheme in time and we can approximate $\dot{E} = \frac{1}{\Delta t}(E^{\tau+1} - E^\tau)$ and $\dot{B} = \frac{1}{\Delta t}(B^{\tau+1} - B^\tau)$

$$(3.15) \iff \begin{cases} \epsilon_0 M \frac{1}{\Delta t}(E^{\tau+1} - E^\tau) - \frac{1}{\mu_0} R B^\tau = 0 \\ \frac{1}{\Delta t}(B^{\tau+1} - B^\tau) - G E^\tau = 0 \end{cases}$$

$$\begin{cases} E^{\tau+1} = E^{\tau} + \frac{1}{\mu_0}\frac{\Delta t}{\epsilon_0} M^{-1} R B^{\tau} \\ B^{\tau+1} = \Delta t G E^{\tau} + B^{\tau} \end{cases} \tag{3.18}$$

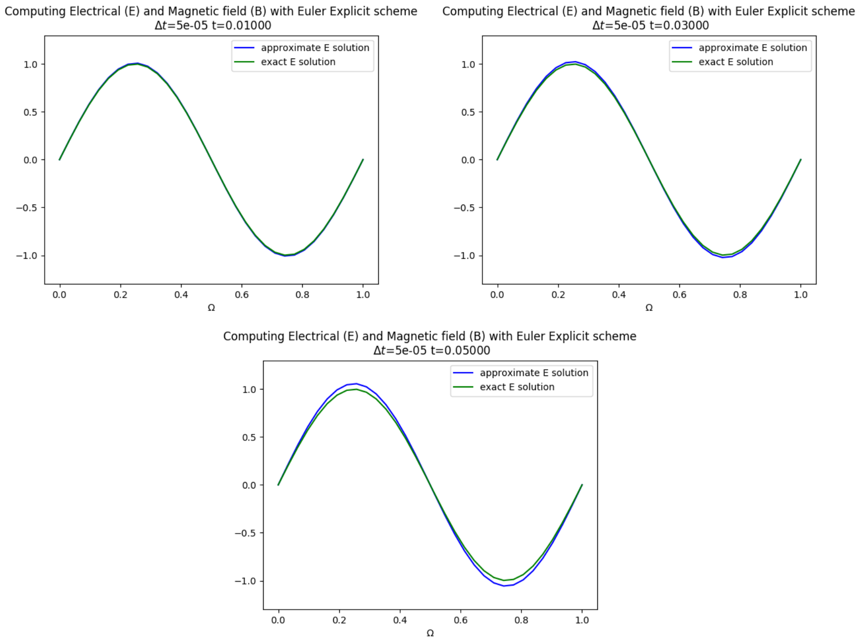

FIGURE 3.3: Numerical results of Electrical Field for Maxwell 1D (Euler Explicit Scheme) with $\Delta t = 5 \times 10^{-5}$

3.5.2 Euler Implicit Scheme

In the same way than before, we approximate $\dot{E}$ and $\dot{B}$

$$(3.15) \iff \begin{cases} \epsilon_0 M \frac{1}{\Delta t}(E^{\tau+1} - E^{\tau}) + \frac{1}{\mu_0} R B^{\tau+1} = 0 \\ \frac{1}{\Delta t}(B^{\tau+1} - B^{\tau}) - G E^{\tau+1} = 0 \end{cases}$$

$$\begin{cases} E^{\tau+1} + \frac{\Delta t}{\epsilon_0 \mu_0} M^{-1} R B^{\tau+1} = E^{\tau} \\ B^{\tau+1} - \Delta t G E^{\tau+1} = B^{\tau} \end{cases} \tag{3.19}$$

We can rewrite the system (3.19) into :

$$\underbrace{\begin{pmatrix} Id_n & \frac{\Delta t}{\epsilon_0 \mu_0} M^{-1} R \\ -\Delta t G & Id_{n-1} \end{pmatrix}}_{=K} \cdot \begin{pmatrix} E^{\tau+1} \\ B^{\tau+1} \end{pmatrix} = \begin{pmatrix} E^{\tau} \\ B^{\tau} \end{pmatrix}$$

and **as K is an inversible matrix**.

Proof. Let's show that K is a inversible matrix.
We know that K is a block matrix and we can define

$$K = \begin{pmatrix} Id_n & \frac{\Delta t}{\epsilon_0 \mu_0} M^{-1} R \\ -\Delta t G & Id_{n-1} \end{pmatrix} = \begin{pmatrix} K_1 & K_2 \\ K_3 & K_4 \end{pmatrix}$$

and we have $K_1 = Id_n$ and $K_4 = Id_{n-1}$ so K_1 and K_4 are square and non-singular, with $K_1^{-1} = K_1$ and $K_4^{-1} = K_4$.
Show that $(K_4 - K_3K_1^{-1}K_2)$ is non-singular:

$$\begin{aligned} K_4 - K_3K_1^{-1}K_2 &= K_4 - K_3K_1K_2 \\ &= Id_{n-1} - (-\Delta t G) \cdot Id_n \cdot \left(\frac{\Delta t}{\mu_0\epsilon_0} M^{-1} R\right) \\ &= Id_{n-1} + \frac{\Delta t^2}{\mu_0\epsilon_0} G M^{-1} R \end{aligned}$$

Finally, we have K a non-singular matrix if $Id_{n-1} + \frac{\Delta t^2}{\mu_0\epsilon_0} G M^{-1} R$ is non-singular. $\square$

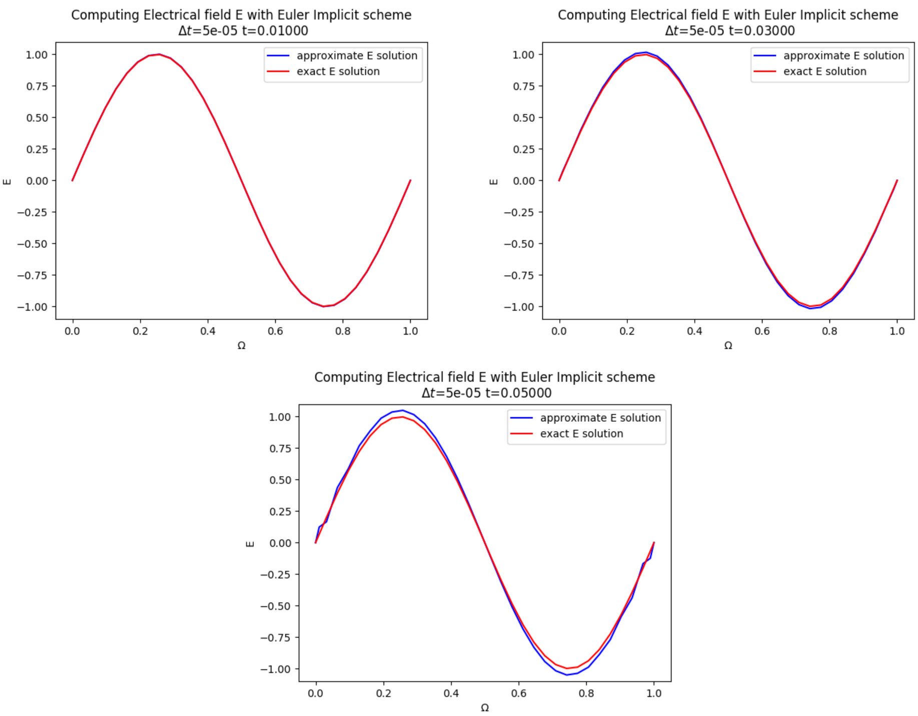

FIGURE 3.4: Numerical results of Electrical Field for Maxwell 1D (Euler Implicit Scheme) with $\Delta t = 5 \times 10^{-5}$

3.5.3 Runge Kutta 4 scheme

We use the Runge Kutta 4 scheme to approximate $\dot{E}$ and $\dot{B}$:

$$(3.15) \iff \begin{cases} E^{\tau+1} = E^{\tau} + \frac{1}{6}(E_{k1} + 2E_{k2} + 2E_{k3} + E_{k4}) \\ B^{\tau+1} = B^{\tau} + \frac{1}{6}(B_{k1} + 2B_{k2} + 2B_{k3} + B_{k4}) \end{cases}$$

with

$$E_{k1} = \Delta t \frac{1}{\mu_0 \epsilon_0} M^{-1} R B^{\tau}$$
$$B_{k1} = \Delta t G E^{\tau}$$
$$E_{k2} = \Delta t \frac{1}{\mu_0 \epsilon_0} M^{-1} R (B^{\tau} + \frac{1}{2} B_{k1})$$
$$B_{k2} = \Delta t G (E^{\tau} + \frac{1}{2} E_{k1})$$
$$E_{k3} = \Delta t \frac{1}{\mu_0 \epsilon_0} M^{-1} R (B^{\tau} + \frac{1}{2} B_{k2})$$
$$B_{k3} = \Delta t G (E^{\tau} + \frac{1}{2} E_{k2})$$
$$E_{k4} = \Delta t \frac{1}{\mu_0 \epsilon_0} M^{-1} R (B^{\tau} + B_{k3})$$
$$B_{k4} = \Delta t G (E^{\tau} + E_{k3})$$

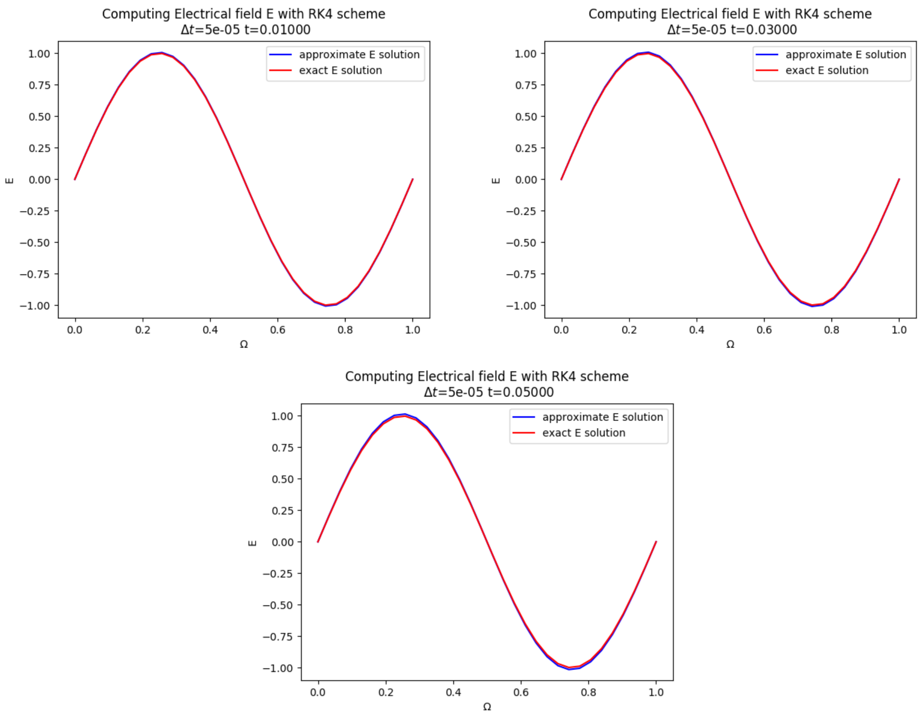

FIGURE 3.5: Numerical results of Electrical Field for Maxwell 1D (Runge Kuta 4 scheme) with $\Delta t = 5 \times 10^{-5}$

3.6 Error of these scheme

We can quickly take a look on the error of each scheme above. As we are working with the H^1 and L^2 spaces, we use the quadratic norm defined

$$err(t) = \sqrt{\int_{\Omega} |f(x,t)|^2 dx}, \quad \forall f \in L^2(\Omega) \tag{3.20}$$

In these cases, we are interesting by computing the error between the values of the exact electric field named $E_{ex}(x,t)$ and the approximate one E. So we compute:

$$\begin{aligned} err_e(t) &= \sqrt{\int_\Omega |E_{ex}(x,t) - E(x,t)|^2 dx} \\ &= \sqrt{\sum_{i=0}^{n_e} \int_{e_i} |E_{ex}(x,t) - E(x,t)|^2 dx} \end{aligned}$$

and we can use the Gauss-Legendre quadrature method with q quadrature points and $\lambda_0, ..., \lambda_{q-1}$ the associated weight, then

$$err_e(t) = \sqrt{\sum_{i=0}^{n_e} \sum_{m=0}^{q-1} \lambda_{m,e} |E_{ex}(x_{m,e},t) - E(x_{m,e},t)|^2}$$

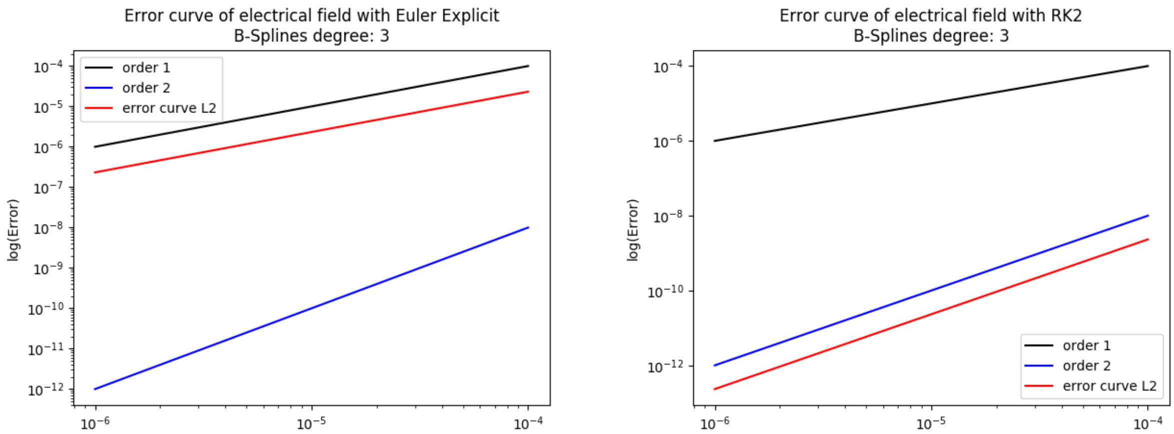

FIGURE 3.6: Order of Euler Explicit and RK2 scheme

On these two figures, we can see that the theorical order of each scheme is verified: the curve of the Euler explicit scheme error is parallel to the curve which represent the first order of error. We can see that the same phenomena exist for the RK2 scheme error curve and the second order error curve.

We can add to this the fact that the degree p of the family of B-Splines interact with the order of the error: at the best the order error of a scheme could be (p-1).

3.7 Energy of the system

The electromagnetic energy is the energy of the electromagnetic field in a volume at a time t. It is defined in Joules (J) in a cubic metre:

$$U_{em}(t) = \frac{1}{2} \int_\Omega (\epsilon_0 \cdot E(x,t)^2 + \frac{1}{\mu_0} \mathcal{B}(x,t)^2) dx \tag{3.21}$$

As we have discretized $E(x,t)$ and $B(x,t)$

- $E(x,t) = \sum_{i=0}^{n} e_i(t) N_i^p(x)$
- $\mathcal{B}(x,t) = \sum_{i=0}^{n-1} b_i(t) N_i^p(x)$

So we can approximate $U_{em}(t)$ at the time t^τ by U^τ:

$$U_{em}(t) = \frac{1}{2}\int_\Omega \epsilon_0 \cdot E(x,t)^2 + \frac{1}{\mu_0}\mathcal{B}(x,t)^2 dx \tag{3.22}$$

$$= \frac{1}{2}\int_\Omega \epsilon_0 E(x,t)^2 dx + \frac{1}{2}\int_\Omega \frac{1}{\mu_0}\mathcal{B}(x,t)^2 dx \tag{3.23}$$

$$\tag{3.24}$$

and we have:

$$\begin{aligned}
\int_\Omega E(x,t)^2 dx &= \int_\Omega (\sum_{i=0}^{n} e_i^\tau N_i^p(x))^2 dx \\
&= \int_\Omega (\sum_{i=0}^{n} e_i^\tau N_i^p(x))(\sum_{j=0}^{n} e_j^\tau N_j^p(x)) dx \\
&= \sum_{i=0}^{n}\sum_{j=0}^{n} e_i^\tau e_j^\tau \int_\Omega N_i^p(x) N_j^p(x) dx \\
&= (E^\tau)^T M E
\end{aligned}$$

In the same way we get:

$$\begin{aligned}
\int_\Omega \mathcal{B}(x,t)^2 dx &= \int_\Omega (\sum_{i=0}^{n-1} b_i^\tau D_i^{p-1}(x))^2 dx \\
&= (\mathcal{B}^\tau)^T M_{\mathcal{B}} B^\tau
\end{aligned}$$

So

$$\begin{aligned}
U_{em}(t) &= \frac{1}{2}\int_\Omega \epsilon_0 \cdot E(x,t)^2 dx + \frac{1}{2}\int_\Omega \frac{1}{\mu_0}\mathcal{B}(x,t)^2 dx \\
\Longrightarrow U^\tau &= \frac{1}{2}\epsilon_0 (E^\tau)^T M E + \frac{1}{2}\frac{1}{\mu_0}(\mathcal{B}^\tau)^T M_{\mathcal{B}} B^\tau
\end{aligned}$$

Chapter 4

Maxwell's equations 2D

4.1 Variational Formulation

We remind the Maxwell's equations (3.5):

$$\begin{cases} -\epsilon_0 \partial_t E + \frac{1}{\mu_0} \nabla \times B = j \\ \partial_t \mathcal{B} + \nabla \times E = 0 \end{cases}$$

and we know that there are two mode for Maxwell's equations in two dimensions: the transverse electric mode (TE) and the transverse magnetic mode (TM).

The transverse electric mode suppose that there is no electric field in the direction of the propagation. We choose here z as the direction of the propagation so we have $E_z = 0$:

$$(3.5) \Longleftrightarrow \begin{cases} \partial_y E_z - \partial_z E_y + \partial_t \mathcal{B}_x = 0 \\ \partial_z E_x - \partial_x E_z + \partial_t \mathcal{B}_y = 0 \\ \partial_x E_y - \partial_y E_x + \partial_t \mathcal{B}_z = 0 \\ -\epsilon_0 \partial_t E_x + \frac{1}{\mu_0}(\partial_y \mathcal{B}_z - \partial_z \mathcal{B}_y) = j_x \\ -\epsilon_0 \partial_t E_y + \frac{1}{\mu_0}(\partial_z \mathcal{B}_x - \partial_x \mathcal{B}_z) = j_y \\ -\epsilon_0 \partial_t E_z + \frac{1}{\mu_0}(\partial_x \mathcal{B}_y - \partial_y \mathcal{B}_x) = j_z \end{cases}$$

$$\Longleftrightarrow \begin{cases} \partial_x E_y - \partial_y E_x + \partial_t \mathcal{B}_z = 0 \\ -\epsilon_0 \partial_t E_x + \frac{1}{\mu_0} \partial_y \mathcal{B}_z = j_x \\ -\epsilon_0 \partial_t E_y - \frac{1}{\mu_0} \partial_x \mathcal{B}_z = j_y \end{cases} \tag{4.1}$$

so we have two different curl operator:

- one acting on scalars we can rename rot : $\nabla \times u = rot(u) = \begin{pmatrix} \partial_y u \\ -\partial_x u \end{pmatrix}$
- one acting on vectors: $u = (u_x, u_y)^T$, $\nabla \times u = curl(u) = \partial_x u_y - \partial_y u_x$

In the same way, we have for the transverse magnetic mode with z the direction of propagation of magnetic field:

$$\begin{cases} \partial_y E_z + \partial_t \mathcal{B}_x = 0 \\ -\partial_x E_z + \partial_t \mathcal{B}_y = 0 \\ -\epsilon_0 \partial_t E_z + \frac{1}{\mu_0}(\partial_x \mathcal{B}_y - \partial_y \mathcal{B}_x = j_z \end{cases} \Longleftrightarrow \begin{cases} \nabla \times E + \partial_t \mathcal{B} = 0 \\ -\epsilon_0 E + \frac{1}{\mu_0} \nabla \times B = j \end{cases} \tag{4.2}$$

where $E = E_z$ and $B = \begin{pmatrix} \mathcal{B}_x \\ \mathcal{B}_y \end{pmatrix}$

In the rest of this part we will interest to the TE mode (4.1).

$$\begin{array}{ccccc} H^1(\Omega) & \xrightarrow{\nabla\times} & \text{H(div}, \Omega) & \xrightarrow{\nabla\cdot} & L^2 \\ \cup & & \cup & & \cup \\ \text{V} & \longrightarrow & W_{div} & \longrightarrow & \text{X} \end{array}$$

FIGURE 4.1: Spaces Representation for Maxwell's equation 2D in TE mode

We consider $\phi \in V_h$ a test function with V_h a test space and Ω a rectangular grid. Then,

$$\int_\Omega \nabla \times E \cdot \phi dx + \int_\Omega \partial_t \mathcal{B} \cdot \phi dx = 0$$

and by using the Green formulation,

$$\int_\Omega g \cdot \nabla \times f dx - \int_{\partial\Omega} (g \times n) \cdot f dS, \forall f \in H(curl, \Omega), \forall g \in H^1(\Omega) \tag{4.3}$$

so

$$\begin{aligned} & \int_\Omega \nabla \times E \cdot \phi dx + \int_\Omega \partial_t \mathcal{B} \cdot \phi dx = 0 \\ \Longleftrightarrow & \int_\Omega E \cdot \nabla \times \phi dx - \int_{\partial\Omega} (E \times n) \cdot \phi dS + \int_\Omega \partial_t \mathcal{B} \cdot \phi dx = 0 \end{aligned}$$

and as we only consider periodic or homogenous boundary conditions, we have $E \times n = 0$ that's why we get

$$\int_\Omega E \cdot \nabla \times \phi dx + \int_\Omega \partial_t \mathcal{B} \cdot \phi dx = 0$$

Finally, we decide to discretize this next system :

$$\begin{cases} \int_\Omega E \cdot \nabla \times \phi dx + \int_\Omega \partial_t \mathcal{B} \cdot \phi dx = 0 \\ \epsilon_0 \partial_t E = \frac{1}{\mu_0} \nabla \times B - j \end{cases} \tag{4.4}$$

4.2 Discretization of the equation

In this part we consider the next spaces where $(N_i^p)_{i \in \{0,\dots,n-1\}}$ is a family of B-Splines of degree p and $(D_i^{p-1})_{\{0,\dots,n-1\}}$ a family of B-Splines of degree p-1[1].

[1] In this part we use the same notations presented into the thesis of Ratnani, 2011

$$
\begin{aligned}
V &= span\{N_i^p(x)N_j^p(y), 0 \le i \le N_x - 1, 0 \le j \le N_y - 1\} \\
W_{div} &= span\{\begin{pmatrix} N_i^p(x)D_j^{p-1}(y) \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ D_i^{p-1}(x)N_j^p(y) \end{pmatrix}, 0 \le i \le N_x - 1, 0 \le j \le N_y - 1\} \\
X &= span\{D_i^{p-1}D_j^{p-1}, 0 \le i \le N_x - 2, 0 \le j \le N_y - 2\}
\end{aligned}
$$

$$
\begin{array}{ccccc}
H^1(\Omega) & \xrightarrow{\nabla\times} & \mathrm{H(div,}\,\Omega) & \xrightarrow{\nabla\cdot} & L^2 \\
\cup & & \cup & & \cup \\
\mathrm{V} & \longrightarrow & W_{div} & \longrightarrow & \mathrm{X}
\end{array}
$$

FIGURE 4.2: Spaces Representation for Maxwell's equation 2D in TE mode

To simplify our notation, we define :

$$
\begin{aligned}
\psi^1_{i_1,i_2} &= \begin{pmatrix} N_{i_1}^p(x)D_{i_2}^{p-1}(y) \\ 0 \end{pmatrix} \\
\psi^2_{i_1,i_2} &= \begin{pmatrix} 0 \\ D_{i_1}^{p-1}(x)N_{i_1}^p(y) \end{pmatrix} \\
\implies W_{div} &= span\{\psi^1_{i_1,i_2}, \psi^2_{i_1,i_2} \quad , \quad 0 \le i_1 \le N_x - 1, 0 \le i_2 \le N_y - 1\}
\end{aligned}
$$

First we take a look to the first equation of (4.4):

$$
\int_\Omega E \cdot \nabla \times \phi dx + \int_\Omega \partial_t \mathcal{B} \cdot \phi dx = 0 \quad \phi \in V_h
$$

As we know $\nabla \times \phi \in W_{div}$ so wa have to choose $V_h = V$, and $E \in W_{div}, \quad \mathcal{B} \in V$

$$
\mathcal{B} = \sum_{i_1=0}^{N_x-1} \sum_{i_2=0}^{N_y-1} b_{i_1,i_2} N_{i_1}^p(x) N_{i_2}^p(y)
$$

and we discretize $E = \begin{pmatrix} E_x \\ E_y \end{pmatrix}$

$$E_x = \sum_{i_1=0}^{N_x-1}\sum_{i_2=0}^{N_y-1} e^x_{i_1,i_2} N^p_{i_1}(x) D^{p-1}_{i_2}(y)$$
$$E_y = \sum_{i_1=0}^{N_x-1}\sum_{i_2=0}^{N_y-1} e^y_{i_1,i_2} D^{p-1}_{i_1}(x) N^p_{i_2}(y)$$
$$\Longrightarrow E = \sum_{i_1=0}^{N_x-1}\sum_{i_2=0}^{N_y-1} e^x_{i_1,i_2}\psi^1_{i_1,i_2} + e^y_{i_1,i_2}\psi^2_{i_1,i_2}$$

We work on the first part of the equation: we have $\nabla \times \phi \in W_{div}$ like $\phi = N_{i_1}(x)N_{i_2}(y), \quad \phi \in V$:

$$\begin{aligned}\nabla \times \phi &= \begin{pmatrix} \partial_y \phi \\ -\partial_x \phi \end{pmatrix} \\ &= \begin{pmatrix} N^p_{i_1}(x)(N^p_{i_2}(x))' \\ -(N^p_{i_1}(x))' N^p_{i_2}(y) \end{pmatrix}\end{aligned}$$

$$\begin{aligned}&\int_\Omega E \cdot \nabla \times \phi dx \\ &= \int_\Omega \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} e^x_{j_1,j_2}\psi^1_{j_1,j_2} + e^y_{j_1,j_2}\psi^2_{j_1,j_2} \cdot \nabla \times \phi dx \\ &= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} \int_\Omega e^x_{j_1,j_2}\psi^1_{j_1,j_2} \cdot \nabla \times \phi dx + \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} \int_\Omega e^y_{j_1,j_2}\psi_{j_1,j_2} \cdot \nabla \times \phi dx \\ &= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} e^x_{j_1,j_2} \int_\Omega N^p_{j_1}(x) D^{p-1}_{j_2}(y) N^p_{i_1}(x)(N^p_{i_2}(x))' dx \\ &+ \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} e^y_{j_1,j_2} \int_\Omega -D^{p-1}_{j_1}(x) N^p_{j_2}(y)(N^p_{i_1}(x))' N^p_{i_2}(x) dx\end{aligned}$$

And we note

$$(R_1)_{i_1,i_2,j_1,j_2} = \int_\Omega N^p_{j_1}(x) D^{p-1}_{j_2}(y) N^p_{i_1}(x)(N^p_{i_2}(x))' dx$$
$$(R_2)_{i_1,i_2,j_1,j_2} = \int_\Omega -D^{p-1}_{j_1}(x) N^p_{j_2}(y)(N^p_{i_1}(x))' N^p_{i_2}(x) dx$$

and we finally define **R the rotational matrix**:

$$R = \begin{bmatrix} R_1 & | & R_2 \end{bmatrix} \tag{4.5}$$

Then we look for the second part of the equation:

$$\int_\Omega \partial_t \mathcal{B} \cdot \phi = \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} \dot{b_{j_1,j_2}} \int_\Omega N^p_{j_1}(x) N_{j_2}(y) N^p_{i_1}(x) N^p_{i_2}(y) dx$$

so the first equation of (4.4) can be discretized by

$$RE + M_v\dot{\mathcal{B}} = 0$$

with M_v **the mass matrix of V space**

We can use the same discretization for the second equation of (4.4):

$$\epsilon_0\partial_t E = \frac{1}{\mu_0}\nabla \times B - j$$

As we have

$$
\begin{aligned}
\mathcal{B} &= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2} N_{j_1}^p(x)N_{j_2}^p(y) \\
\Longleftrightarrow \nabla \times \mathcal{B} &= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2} \begin{pmatrix} N_{j_1}^p(x)(D_{j_2}^{p-1}(y) - D_{j_2+1}^{p-1}(y)) \\ -(D_{j_1}^{p-1}(x) - D_{j_1+1}^{p-1}(x))N_{j_2}^p(y) \end{pmatrix} \\
&= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}(\psi^1_{j_1,j_2} - \psi^1_{j_1,j_2+1} - \psi^2_{j_1,j_2} + \psi^2_{j_1+1,j_2}) \\
&= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}(\psi^1_{j_1,j_2} - \psi^1_{j_1,j_2+1}) + \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}(-\psi^2_{j_1,j_2} + \psi^2_{j_1+1,j_2}) \\
&= \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}\psi^1_{j_1,j_2} - \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}\psi^1_{j_1,j_2+1} - \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}\psi^2_{j_1,j_2} + \sum_{j_1=0}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}\psi^2_{j_1+1,j_2} \\
&= \sum_{j_1=0}^{N_x-1} b_{j_1,0}\psi^1_{j_1,0} + \sum_{j_1=0}^{N_x-1}\sum_{j_2=1}^{N_y-1} b_{j_1,j_2}\psi^1_{j_1,j_2} - \sum_{j_1=0}^{N_x-1}\sum_{j_2=1}^{N_y-1} b_{j_1,j_2-1}\psi^1_{j_1,j_2} + \sum_{j_1=0}^{N_x-1} b_{j_1,N_y-1}\underbrace{\psi^1_{j_1,N_y}}_{=0} \\
&\quad - \sum_{j_2=0}^{N_y-1} b_{0,j_2}\psi^2_{0,j_2} - \sum_{j_1=1}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1,j_2}\psi^2_{j_1,j_2} + \sum_{j_1=1}^{N_x-1}\sum_{j_2=0}^{N_y-1} b_{j_1-1,j_2}\psi^2_{j_1,j_2} + \sum_{j_2=0}^{N_y-1} b_{N_x-1,j_2}\underbrace{\psi^2_{N_x,j_2}}_{=0} \\
&= \sum_{j_1=1}^{N_x-1}\sum_{j_2=1}^{N_y-1}(b_{j_1,j_2} - b_{j_1,j_2-1})\psi^1_{j_1,j_2} - \sum_{j_1=1}^{N_x-1}\sum_{j_2=1}^{N_y-1}(b_{j_1,j_2} - b_{j_1+1,j_2})\psi^2_{j_1,j_2} + \sum_{j_1=0}^{N_x-1} b_{j_1,0}\psi^1_{j_1,0} - \sum_{j_2=0}^{N_y-1} b_{0,j_2}\psi_{0,j_2} \\
&=(\sum_{j_1=0}^{N_x-1} b_{j_1,0}\psi^1_{j_1,0} + \sum_{j_1=0}^{N_x-1}\sum_{j_2=1}^{N_y-1}(b_{j_1,j_2} - b_{j_1,j_2-1})\psi^1_{j_1,j_2}) - (\sum_{j_1=1}^{N_x-1}\sum_{j_2=0}^{N_y-1}(b_{j_1,j_2} - b_{j_1-1,j_2})\psi^2_{j_1,j_2} + \sum_{j_2=0}^{N_y-1} b_{0,j_2}\psi^2_{0,j_2})
\end{aligned}
$$

So we have

$$\nabla \times \mathcal{B} = \underbrace{\begin{pmatrix} G_1 \otimes I_{N_y} \\ -I_{N_x} \otimes G_2 \end{pmatrix}}_{=\mathbb{G}} B$$

$$G_1 = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathcal{M}_{N_x-1,N_x}(\mathbb{R})$$

$$G_2 = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & -1 & 1 \end{pmatrix} \in \mathcal{M}_{N_y-1,N_y}(\mathbb{R})$$

and the second equation of (4.4) can be discretized as:

$$\epsilon_0 \dot{E} = \frac{1}{\mu_0} \mathbb{G} B - J$$

Finally we have the next system, which discretize our problem:

$$\begin{cases} \dot{E} = \frac{1}{\mu_0 \epsilon_0} \mathbb{G} B - \frac{1}{\epsilon_0} J \\ \dot{\mathcal{B}} = -(M_v)^{-1} R E \end{cases} \tag{4.6}$$

4.3 Some examples of scheme:

In this part we will referred to the next analytical solution:

$$\begin{aligned} H(x,y,t) &= \cos(k_1 x + \sigma_1) \sin(k_2 y + \sigma_2) \cos(\omega t) \\ E_x(x,y,t) &= -\frac{k_2}{\omega} \cos(k_1 x + \sigma_1) \sin(k_2 y + \sigma_2) \sin(\omega t) \\ E_y(x,y,t) &= \frac{k_2}{\omega} \sin(k_1 x + \sigma_1) \cos(k_2 y + \sigma_2) \sin(\omega t) \end{aligned}$$

and we suppose: j = 0, $k_1 = k_2 = 1$, $\sigma_1 = \sigma_2 = 0$ and $\omega = \pi$

4.3.1 Euler Explicit:

We will approximate $\dot{E}$ and $\dot{B}$: $\dot{E} = \frac{1}{\Delta t}(E^{\tau+1} - E^{\tau})$ and $\dot{B} = \frac{1}{\Delta t}(E^{\tau+1} - E^{\tau})$ so we can rewrite the (4.6) system:

$$(4.6) \Longleftrightarrow \begin{cases} E^{\tau+1} = E^{\tau} + \frac{\Delta t}{\mu_0 \epsilon_0} G B^{\tau} \\ B^{\tau+1} = B^{\tau} - M_v^{-1} R E^{\tau} \end{cases}$$

4.3.2 Runge Kutta 4:

In this section, we choose to approximate E and B by using the Runge Kutta 4 scheme:

$$(4.6) \Longleftrightarrow \begin{cases} E^{\tau+1} = E^{\tau} + \frac{1}{6}(E_{k1} + 2E_{k2} + 2E_{k3} + E_{k4}) \\ B^{\tau+1} = B^{\tau} + \frac{1}{6}(B_{k1} + 2B_{k2} + 2B_{k3} + B_{k4}) \end{cases}$$

with

$$\begin{aligned}
E_{k1} &= \Delta t \frac{1}{\mu_0 \epsilon_0} G B^{\tau} \\
B_{k1} &= -\Delta t (M_v)^{-1} R E^{\tau} \\
E_{k2} &= \Delta t \frac{1}{\mu_0 \epsilon_0} G (B^{\tau} + \frac{1}{2} B_{k1}) \\
B_{k2} &= -\Delta t (M_v)^{-1} R (E^{\tau} + \frac{E_{k1}}{2}) \\
E_{k3} &= \Delta t \frac{1}{\mu_0 \epsilon_0} G (B^{\tau} + \frac{1}{2} B_{k2}) \\
B_{k3} &= -\Delta t (M_v)^{-1} R (E^{\tau} + \frac{1}{2} E_{k2}) \\
E_{k4} &= \Delta t \frac{1}{\mu_0 \epsilon_0} G (B^{\tau} + B_{k3}) \\
B_{k4} &= -\Delta t (M_v)^{-1} R (E^{\tau} + E_{k3})
\end{aligned}$$

Conclusion

In this report, we have studied Maxwell's equations in one and two dimensions (which means that the wave is moving into one or two dimensions) and the approximation of these equations with B-Splines interpolation. So, we have learned the properties of a family of B-splines and Bernstein polynomials, and how to use them into a numerical method. Besides the theoretical results presented here, we have programmed Maxwell's equation resolution into one and two directions, just like the computational energy of the system in each case. Some of these programs are not finished yet but I really hope to finish them and continue to work on this subject which is very interesting.

This work allowed the team of the NMPP a new eye and maybe slightly naïve on the code of the module SPL. Furthermore, my work is using the SPL module, so it could use as an example and will help to improve the SPL module. I myself believe I learned a lot during this internship, in particular, B-Splines interpolation method. Furthermore, more than improving my knowledge of the Python programming language, I have improved my knowledge of the finite element method. I also wish to remember how we work at the IPP because the way we work in a research institute is rather different from the way we work in a company, where time constraints are not the same just like the work atmosphere.

Unfortunately, due to some practical reason, I didn't progress as much as I wanted into Maxwell's equations 2D properties, where I would like to study the energy of the system and the different projectors of the transverse electric mode.

Appendix A

Notation of Gradient, curl and divergence operator

We use differential operators where the notation using the nabla operator ∇ is used: $\nabla = (\partial_1, \partial_2, \partial_3)^T$

- **gradient operator**:

$$\vec{\vec{\nabla}} u = \begin{pmatrix} \partial_1 u \\ \partial_2 u \\ \partial_3 u \end{pmatrix}$$

- **curl operator**:

$$\nabla \times u = \begin{pmatrix} \partial_2 u_3 - \partial_3 u_2 \\ \partial_3 u_1 - \partial_1 u_3 \\ \partial_1 u_2 - \partial_2 u_1 \end{pmatrix}$$

- **divergence operator**:

$$\nabla \cdot u = \sum_{i=1}^{3} \partial_i u_i$$

We can also remember some properties of these operators:

- $\forall P \in \mathbb{R}, \nabla \cdot (\vec{\vec{\nabla}} P) = \Delta P$
- $\forall P \in \mathbb{R}^3, \nabla \cdot (\nabla \times P) = 0$
- $\forall P \in \mathbb{R}^3, \nabla \times (\nabla \times P) = \vec{\vec{\nabla}} (\nabla \cdot P) - \Delta P$

Proof.
- $\underline{\forall P \in \mathbb{R}, \nabla \cdot (\vec{\vec{\nabla}} P) = \Delta P}$:
we consider $P \in \mathbb{R}$:

$$\vec{\vec{\nabla}} P = \begin{pmatrix} \partial_1 P \\ \partial_2 P \\ \partial_3 P \end{pmatrix}$$

so

$$\nabla \cdot \vec{\vec{\nabla}} P = \partial_1^2 P + \partial_2^2 P + \partial_3^2 P = \Delta P$$

- $\underline{\forall P \in \mathbb{R}^3, \nabla \cdot (\nabla \times P) = 0}$:
  $P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix}$
  and
  $$\nabla \times P = \begin{pmatrix} \partial_2 P_3 - \partial_3 P_2 \\ \partial_3 P_1 - \partial_1 P_3 \\ \partial_1 P_2 - \partial_2 P_1 \end{pmatrix}$$
  so
  $$\begin{aligned} \nabla \cdot \nabla \times P &= \partial_{21} P_3 - \partial_{31} P_2 + \partial_{32} P_1 - \partial_{12} P_3 + \partial_{13} P_2 - \partial_{23} P_1 \\ &= 0 \end{aligned}$$

$\square$

Appendix B

Gauss-Legendre quadrature method

The aim of Gauss-Legendre quadrature method is to approximate the integral of a function. It is most of time define on the [-1;1] interval. We will discretize Ω with some "quadrature points" q_i, $i \in \{1, ..., n\}$.

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{n} \lambda_i f(q_i) \tag{B.1}$$

and we note λ_i weight of the quadrature point q_i.

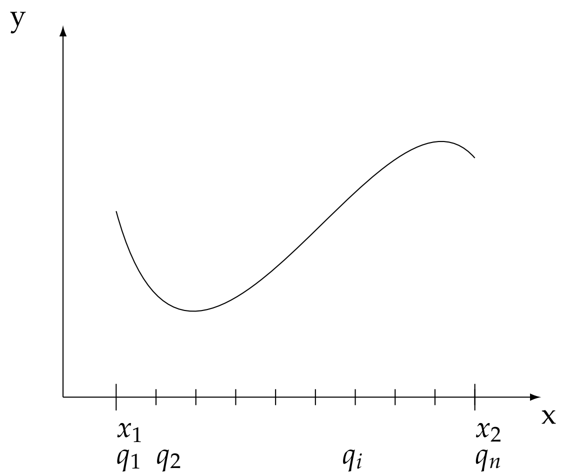

FIGURE B.1: Illustration of the Legendre-Gauss quadrature method

If we want tot generalize this method to a [a;b] interval , $a, b \in \mathbb{R}$ then we use a change of variable:

$$\begin{aligned}
\int_{-1}^{1} f(x)dx &\underset{y=x+1}{=} \int_{0}^{2} f(y)dy \\
&\underset{y=\frac{y}{2}}{=} \int_{0}^{1} \frac{1}{2} f(y)dy \\
&\underset{y=y(b-a)}{=} \int_{0}^{b-a} \frac{1}{2}(b-a) f(y)dy \\
&\underset{y=y+a}{=} \int_{a}^{b} \frac{1}{2}(b-a) f(y)dy
\end{aligned}$$

so we have

$$\begin{aligned}\int_a^b f(x)dx &= \int_{-1}^1 \frac{b-a}{2} f(\frac{b-a}{2}x + \frac{b+a}{2}) \\ &= \frac{b-a}{2}\sum_{i=1}^n \lambda_i f(\frac{b-a}{2}q_i + \frac{b+a}{2})\end{aligned}$$

Appendix C

The Kronecker Product

The Kronecker product[1], represented by the symbol $\otimes$ is defined for $A \in \mathcal{M}_2(\mathbb{R})$ and $B \in \mathcal{M}_3(\mathbb{R})$

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}}_{=A} \otimes \underbrace{\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}}_{=B} = \begin{pmatrix} \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & a_{11}b_{13} \\ a_{11}b_{21} & a_{11}b_{22} & a_{11}b_{23} \\ a_{11}b_{31} & a_{11}b_{32} & a_{11}b_{33} \end{bmatrix} & \begin{bmatrix} a_{12}b_{11} & a_{12}b_{12} & a_{12}b_{13} \\ a_{12}b_{21} & a_{12}b_{22} & a_{12}b_{23} \\ a_{12}b_{31} & a_{12}b_{32} & a_{12}b_{33} \end{bmatrix} \\ \begin{bmatrix} a_{21}b_{11} & a_{21}b_{12} & a_{21}b_{13} \\ a_{21}b_{21} & a_{21}b_{22} & a_{21}b_{23} \\ a_{21}b_{31} & a_{21}b_{32} & a_{21}b_{33} \end{bmatrix} & \begin{bmatrix} a_{22}b_{11} & a_{22}b_{12} & a_{22}b_{13} \\ a_{22}b_{21} & a_{22}b_{22} & a_{22}b_{23} \\ a_{22}b_{31} & a_{22}b_{32} & a_{22}b_{33} \end{bmatrix} \end{pmatrix} \tag{C.1}$$

C.1 Basic properties of the Kronecker Product

Property C.1.1. Bilinearity and associativiy of the Kronecker Product
If we consider A, B, C three matrices and we suppose their shape are compatible then, $\forall \lambda \in \mathbb{R}$

1. $A \otimes (B + \lambda C) = (A \otimes B) + \lambda(A \otimes C)$

   Proof. we consider $B, C \in \mathcal{M}_{p,q}(\mathbb{R})$ and $A \in \mathcal{M}_{n,m}(\mathbb{R})$,

   so

$$\begin{aligned} A \otimes (B + \lambda C) &= \begin{pmatrix} a_{11}(B+\lambda C) & \dots & a_{m1}(B+\lambda C) \\ \dots & a_{ij}(B+\lambda C) & \dots \\ a_{n1}(B+\lambda C) & \dots & a_{nm}(B+\lambda C) \end{pmatrix} \in \mathcal{M}_{np,mq}(\mathbb{R}) \\ &= \begin{pmatrix} a_{11}B & \dots & a_{m1}B \\ \dots & a_{ij}B & \dots \\ a_{n1}B & \dots & a_{nm}B \end{pmatrix} + \begin{pmatrix} a_{11}\lambda C & \dots & a_{m1}\lambda C \\ \dots & a_{ij}\lambda C & \dots \\ a_{n1}\lambda C & \dots & a_{nm}\lambda C \end{pmatrix} \\ &= (A \otimes B) + \lambda(A \otimes C) \end{aligned}$$

   $\square$

2. $(A + \lambda B) \otimes C = (A \otimes C) + \lambda(B \otimes C)$

[1] cf. the article of Loan, 1999

Proof. In the same way than before, we choose here $A, B \in \mathcal{M}_{n,m}(\mathbb{R})$ and $C \in \mathcal{M}_{p,q}(\mathbb{R})$

$$\begin{aligned}
(A + \lambda B) \otimes C &= \begin{pmatrix} (a_{11} + \lambda b_{11})C & ... & (a_{1m} + \lambda b_{1m})C \\ ... & (a_{ij} + \lambda b_{ij})C & ... \\ (a_{n1} + \lambda b_{n1})C & ... & (a_{nm} + \lambda b_{nm})C \end{pmatrix} \\
&= \begin{pmatrix} a_{11}C & ... & a_{1m}C \\ ... & a_{ij}C & ... \\ a_{n1}C & ... & a_{nm}C \end{pmatrix} + \begin{pmatrix} \lambda b_{11}C & ... & \lambda b_{1m}C \\ ... & \lambda b_{ij}C & ... \\ \lambda b_{n1}C & ... & \lambda b_{nm}C \end{pmatrix} \\
&= (A \otimes C) + \lambda(B \otimes C)
\end{aligned}$$

□

3. $A \otimes (B \otimes C) = (A \otimes B) \otimes C$

Proof. We have $A \in \mathcal{M}_{p,q}(\mathbb{R})$ and $B \in \mathcal{M}_{n,m}(\mathbb{R})$

$$B \otimes C = \begin{pmatrix} b_{11}C & ... & b_{1m}C \\ ... & ... & ... \\ b_{n1}C & ... & b_{nm}C \end{pmatrix}$$

and so

$$\begin{aligned}
A \otimes (B \otimes C) &= \begin{pmatrix} \left[a_{11} \begin{pmatrix} b_{11}C & ... & b_{1m}C \\ ... & ... & ... \\ b_{n1}C & ... & b_{nm}C \end{pmatrix} \right] & ... & \left[a_{1q} \begin{pmatrix} b_{11}C & ... & b_{1m}C \\ ... & ... & ... \\ b_{n1}C & ... & b_{nm}C \end{pmatrix} \right] \\ ... & ... & ... \\ \left[a_{p1} \begin{pmatrix} b_{11}C & ... & b_{1m}C \\ ... & ... & ... \\ b_{n1}C & ... & b_{nm}C \end{pmatrix} \right] & ... & \left[a_{pq} \begin{pmatrix} b_{11}C & ... & b_{1m}C \\ ... & ... & ... \\ b_{n1}C & ... & b_{nm}C \end{pmatrix} \right] \end{pmatrix} \\
&= \begin{pmatrix} \left[a_{11} \begin{pmatrix} b_{11} & ... & b_{1m} \\ ... & ... & ... \\ b_{n1} & ... & b_{nm} \end{pmatrix} \right] & ... & \left[a_{1q} \begin{pmatrix} b_{11} & ... & b_{1m} \\ ... & ... & ... \\ b_{n1} & ... & b_{nm} \end{pmatrix} \right] \\ ... & ... & ... \\ \left[a_{p1} \begin{pmatrix} b_{11} & ... & b_{1m} \\ ... & ... & ... \\ b_{n1} & ... & b_{nm} \end{pmatrix} \right] & ... & \left[a_{pq} \begin{pmatrix} b_{11} & ... & b_{1m} \\ ... & ... & ... \\ b_{n1} & ... & b_{nm} \end{pmatrix} \right] \end{pmatrix} \otimes C \\
&= (A \otimes B) \otimes C
\end{aligned}$$

□

Property C.1.2. We consider $B \in \mathcal{M}_{n \times m}(R)$ and $C \in \mathcal{M}_{p \times q}(R)$

1. $(B \otimes C)^T = B^T \otimes C^T$

Proof. We start by defining $(B \otimes C)_{ij} = b_{\alpha\beta}c_{\gamma\delta}$, where $\alpha \in \{1, ..., n\}$, $\beta \in \{1, ..., m\}$, $\gamma \in \{1, ..., p\}$ and $\delta \in \{1, ..., q\}$.

And we explain i and j with $\alpha, \beta, \gamma \quad and \quad \delta$ by using **Euclidian decomposition**:

$$i = (\alpha - 1) \times p + \gamma$$
$$j = (\beta - 1) \times q + \delta$$

In this case, $\forall i \in \{1, ..., np\}, \forall j \in \{1, ..., mq\}$

$$\begin{aligned}(B \otimes C)^T_{ij} &= (B \otimes C)_{ji} \\ &= b_{\beta\alpha} \cdot c_{\delta\gamma} \\ &= b^T_{\alpha\beta} \cdot c^T_{\gamma\delta} \\ &= B^T \otimes C^T\end{aligned}$$

□

2. $(B \otimes C)^{-1} = B^{-1} \otimes C^{-1}$

3. $tr(B \otimes C) = tr(A) \otimes tr(B)$

C.2 Algorithm

Algorithm 3 Compute the Kronecker product between A and B matrices : $A \otimes B$ (Python language)

Require: $A \in \mathcal{M}_{m_1,m_2}(\mathbb{R})$, $B \in \mathcal{M}_{n_1,n_2}(\mathbb{R})$
Ensure: $K = A \otimes B$

```
K ← ndarray((m1, m2, n1, n2))
for i ∈ {0, ..., m1 − 1} do
    for j ∈ {0, ..., m2 − 1} do
        K[i, j, :, :] ← A[i, j] · B
    end for
end for
return K
```

Appendix D

Spaces Definitions and properties

The definitions and properties below result from the thesis of Schenk, 2016.

D.1 The Schoenberg space

We note $\mathbb{S}^k$ the Schoenberg space whose canonical basis is the B-Splines family $B_{i,k}$, $\forall i \in \{0, ..., m-1\}$

D.2 The H(curl, Ω) space

Definition D.2.1. The vector space H(curl, Ω) is for a bounded, 3D Lipschitz domain Ω given by:

$$H(curl, \Omega) = \{v \in L^2(\Omega) | \nabla \times v \in L^2(\Omega)\}$$

Definition D.2.2. The H(curl, Ω) norm is defined for $v \in H(curl, \Omega)$ by:

$$\|v\|_{H(curl,\Omega)} = \left(\|v\|^2_{L^2(\Omega)} + \|\nabla \times v\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}$$

Definition D.2.3. The H(curl, Ω) inner product is given for $u, v \in H(curl, \Omega)$ by:

$$(u, v)_{H(curl,\Omega)} = (u, v)_{L^2(\Omega)} + (\nabla \times u, \nabla \times v)_{L^2(\Omega)}$$

D.3 The H(div, Ω) space

Definition D.3.1. The vector space H(div, Ω) is for a bounded, 3D Lipschitz domain Ω given by:

$$H(div, \Omega) = \{v \in L^2(\Omega) | \nabla \cdot v \in L^2(\Omega)\}$$

Definition D.3.2. The H(div, Ω) norm is defined for $v \in H(div, \Omega)$ by:

$$\|v\|_{H(div,\Omega)} = \left(\|v\|^2_{L^2(\Omega)} + \|\nabla \cdot v\|^2_{L^2(\Omega)}\right)^{\frac{1}{2}}$$

Definition D.3.3. The H(div, Ω) inner product is given for $u, v \in H(div, \Omega)$ by:

$$(u, v)_{H(div,\Omega)} = (u, v)_{L^2(\Omega)} + (\nabla \cdot u, \nabla \cdot v)_{L^2(\Omega)}$$

Appendix E

Some Definitions and Properties

E.1 electromagnetic wave

An electromagnetic wave consists of an electric field E and a magnetic (most of the time they are named E and $\mathcal{B}$) oscillating with the same frequency. We know that both fields are perpendicular one to another.

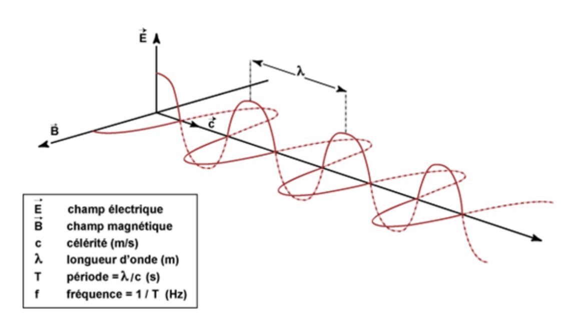

FIGURE E.1: Description of an electromagnetic wave

E.2 Spline Definition

a **spline** is a piecewise by polynomials. In interpolation problems, spline interpolation referred to polynomial interpolation.

E.3 Greville abscissae

We call **Greville abscissae** or **nodes** the average of the knots. We can define them for a knot list $\chi_0, ..., \chi_{m-1}$ and a family of B-Splines $(N_i^p)_{i\in\{0,...,m-p-1\}}$

$$\overline{x}_i = \frac{1}{p-1}\sum_{k=0}^{p-1} \chi_{i+k}$$

E.4 Block matrix inversion

We consider A, B, C and D matrices as $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$

M is non-singular if

- A and D are square matrices

- $D - CA^{-1}B$ and A and D have to be non-singular

and we have:

$$M^{-1} = \begin{pmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -D^{-1}C(A - BD^{-1}C)^{-1} & D^{-1} + D^{-1}C(A - BD^{-1}C)^{-1}BD^{-1} \end{pmatrix} \tag{E.1}$$

Bibliography

Basdevant, C. (2012). “Courbes pour la CAO, Courbes de Bézier et B-Splines”. In: cours Licence, Université Paris13.

Bossavit, Alain (1998). “Computational Electromagnetism - Variational Formulations, Complementarity, Edge Elements”. In: Academic Press. Chap. Introduction: Maxwell Equations.

Loan, Charles F. Van (1999). “The ubiquitous Kronecker product”. In: *Journal of Computational and Applied Mathematics 123 (2000) 85-100*. Department of Computer Science, Cornell Universityy, Ithaca, New York, USA.

Manni, Carla et al. (unpublished yet). “Auxiliary Splines Space Preconditioning for H(curl, Ω) and H(div, Ω) problems”. In:

Ratnani, Ahmed (2011). “Isogeometric Analysis in Plasma Physics and Electromagnetism”. PhD thesis. Université de Strasbourg.

Sanchez-Palencia, Evariste (2013). “Maxwell, Hertz et l’électromagnétisme”. In: *Évolution des disciplines et histoire des découvertes*. Histoire des sciences - Académie des sciences.

Schenk, Benedikt (2016). “Isogeometric Analysis for Electromagnetic Waves in Magnetized Tokamak Plasma”. PhD thesis. TUM, department of Mathematics.