MIMESIS

UNIVERSITY OF STRASBOURG

MASTER CSMI

Internship Report : Innovative non-conformal finite element methods for augmented surgery

Authors:
Frédérique Lecourtier

Supervisors:
Michel Duprez
Emmanuel Franck
Vanessa Lleras

Inria

Date: August 21, 2023

Contents

1 Introduction

This end of study internship is a 2nd year internship in the CSMI Master ("Calcul Scientifique et Mathématiques de l'Information") of the University of Strasbourg. It is the continuation of a project done during the first semester of M2, the main objective of this project was the discovery of an innovative non-conformal finite element method for augmented surgery, the ϕ-FEM method. The purpose of this project was to have Cemosis and Mimesis collaborate through the use of Feel++ software (developed by Cemosis) in the framework of the ϕ-FEM method (one of the research topics of the Mimesis team). This project was followed by a 6-month internship whose main objective was to correct the output of a Fourier Neural Operator (FNO) by a solver using the ϕ-FEM method.

1.1 Scientific Context

Finite element methods (FEM) are used to solve partial differential equations numerically. These can, for example, represent analytically the dynamic behavior of certain physical systems (mechanical, thermodynamic, acoustic, etc.). Among other things, it is a discrete algorithm for determining the approximate solution of a partial differential equation (PDE) on a compact domain with boundary conditions.
The standard FEM method, which requires precise meshing of the domain under consideration and, in particular, fitting with its boundary, has its limitations. In particular, in the medical field, meshing complex and evolving geometries such as organs (e.g. the liver) can be very costly. More specifically, in the application context of creating real-time digital twins of an organ, the standard FEM method would require complete remeshing of the organ each time it is deformed, which in practice is not workable.
This is why other methods, known as non-conformal finite element methods, have emerged in the last few years. These include CutFEM [5] or XFEM [15], based on the idea of introducing a fictitious domain larger than the domain under consideration. We're interested here in another non-conformal method, which we'll present in more detail later, called ϕ-FEM. We'll only use it in the context of Poisson problem solving, for Dirichlet boundary conditions [9]. But the method has been extended to Neumann conditions [8] and then to solve various mechanical problems, including linear elasticity [6, Chapter 2] and heat transfer problems [6, Chapter 5].

1.2 Presentation of the team

Created in January 2021 within ICube laboratory at the University of Strasbourg, MLMS[1] ("Machine Learning, Modélisation et Simulation") team is interested in data, models and simulations for medical science and human motion. It brings together computer scientists, mathematicians, bio-mechanicians, and neuroscientists to develop functional, physical, and

[1] MLMS : https://mlms.icube.unistra.fr/en/index.php/Presentation

geometric models around a transverse axis "Assistance to medical interventions by computer". MLMS hosts the MIMESIS[2] project-team as a sub-team. The MIMESIS research team aims at creating real-time digital twins of an organ, with main application domains as surgical training and surgical guidance during complex interventions. In 2023, a new inria team NECTARINE will be created within MLMS, who will focus on scientific challenges related to neuro-stimulation in the clinical context.
MIMESIS, directed by Stéphane Cotin, is a joint Inria[3] ("Institut national de recherche en sciences et technologies du numérique") and CNRS[4] ("Centre national de la recherche scientifique") Research Team. The Mimesis research team is working on a set of scientific challenges in scientific computing, data assimilation, machine learning and control, with the goal of creating real-time digital twins of an organ.

1.3 Objectives

The main objective of the internship was to combine finite element methods and Machine Learning in order to solve the Poisson problem with Dirichlet condition. More precisely, we want to train a neural network called Fourier Neural Network (FNO) [11] to predict the solutions of a PDE for a given problem family (i.e. a "type" of source term). This neural network is trained with a data set consisting of the ϕ-FEM solutions of the problems considered. The predictions of this neural network will then be fed back into a finite element solver to apply a correction to improve the accuracy of the solution : this was the subject covered during the internship. The finite element methods considered will be presented in Section 2 and the FNO in Section 3.
It is important to note that the ϕ-FEM method has an advantage that is very interesting in the context of organ geometries. Indeed, this type of geometry can deform in time and meshing a fictitious domain around this geometry avoids having to remesh the geometry in time. Thus only the level-set function will be modified and the mesh can be fixed. Moreover, a Cartesian mesh of the fictitious domain allows us to use the same type of neural network as those applied to images (especially FNO).
To be more precise, we will test different correction methods (presented in Section 4.2) on different problems (presented in Section 4.1) which will enable us to use the network prediction to help the solver get as close as possible to the solution. We will start by testing these different types of solver on an analytical solution (Section 4.4.1), then on a "manually perturbed" solution (Section 4.4.2) and finally on a ϕ-FEM solution (Section 4.4.3).
After testing the various types of correction on the previous test cases, we'll apply these same methods to the prediction of an FNO (Section 4.4.4). The main objective is to enable the combination of FNO and correction to be more accurate than the conventional ϕ-FEM solver. By first testing the different corrections on the previous test cases, we hope to get an idea of the order of errors to be expected. During the course of the internship, we realized that the

[2]MIMESIS : https://mimesis.inria.fr/
[3]Inria : https://www.inria.fr/fr
[4]CNRS : https://www.cnrs.fr/fr

results obtained on the FNO did not correspond to the expected analytical results. For this reason, other types of neural networks were considered, namely multi-perceptron networks (Section 4.4.5.1) and PINNs (Section 4.4.5.2), with the aim of checking whether the results obtained are related to the use of the FNO.

1.4 Deliverables

In the context of the internship, the following deliverables are provided:

- a weekly tracking report, written in French, was produced as the internship progressed, listing the objectives and results for each week.
- a github repository containing all the code allowing to reproduce the results presented in this report, as well as the documents written during the internship. The codes have been implemented in Python: for the finite element solvers, we'll be using the FEniCS library, and for the neural network implementation, we'll be using Tensorflow and Pytorch.
- an online report generated with a tool called antora[5]. A continuous integration has been set up on github to execute a python code for each new push, enabling the latex file to be converted directly into this antora documentation.

[5] Antora : https://antora.org/

2 Finite Element Methods (FEMs)

In the following, we will consider the Poisson problem with Dirichlet condition (homogeneous or non-homogeneous):
Problem : Find $u : \Omega \to \mathbb{R}^d$ such that

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega, \end{cases}$$

with Δ the Laplace operator and $\Omega \subset \mathbb{R}^d$ a lipschitzian bounded open set (and $\partial\Omega$ its boundary).
Here, we present the 2 finite element methods we will be considering. First, we will present the standard FEM method in Section 2.1, followed by the ϕ-FEM method in Section 2.2.

2.1 Standard FEM

In this section, we will present the standard finite element method. We'll start by presenting some general notions of functional analysis, then explain the general principle of FEM. Then we'll give a few more details on the method and finish by describing the application to the Poisson problem (with Dirichlet condition). For more information, please refer to [16] and [14].

2.1.1 Some notions of functional analysis.

In this section, we'll recall some of the notions of functional analysis that will be used in the next sections. In particular, Lebesgue spaces and Sobolev spaces. Please refer to the book [3].
Let's consider Ω an open of $\mathbb{R}^d$ ($d = 1,2,3$) with boundary Γ.
We begin here by defining Lebesgue spaces:

Definition 2.1 (Lebesgue spaces)**.** *Lebesgue spaces, denoted L^p, are vector spaces of classes of functions whose exponent power p is integrable in the Lebesgue sense, where p is a strictly positive real number. They are defined by*

$$L^p(\Omega) = \left\{ u : \Omega \to \mathbb{R} | \int_\Omega u^p \, dv < +\infty \right\}$$

In particular, taking $p = 2$, we define the space

$$L^2(\Omega) = \left\{ u : \Omega \to \mathbb{R} | \int_\Omega u^2 \, dv < +\infty \right\}$$

which is the space of integrable square functions.

We also define Sobolev spaces of order 1 and order 2:

Definition 2.2 (Sobolev spaces)**.** *The Sobolev space of order 1, denoted H^1, is defined by*

$$\begin{aligned} H^1(\Omega) &= \left\{u \in L^2(\Omega) | \partial_{x_i} u \in L^2(\Omega)\right\} \\ &= \left\{u \in L^2(\Omega), \nabla u \in L^2(\Omega)^d\right\} \end{aligned}$$

with the scalar product $\langle u, v\rangle_{H^1(\Omega)}$, defined by :

$$\langle u, v\rangle_{H^1(\Omega)} = \int_\Omega uv + \nabla u \cdot \nabla v, \forall u, v \in Hq(\Omega)$$

and the induced norm $||\cdot||_{H^1(\Omega)}$.
We also define the space

$$H_0^1(\Omega) = \left\{u \in H^1(\Omega) | u_{|\Gamma} = 0\right\}$$

The Sobolev space of order 2, denoted H^2, is defined by

$$H^2(\Omega) = \left\{u, u', u'' \in L^2(\Omega)\right\}$$

with scalar product $\langle u, v\rangle_{H^2(\Omega)}$, defined by :

$$\langle u, v\rangle_{H^2(\Omega)} = \int_\Omega uv + u'v' + u''v'', \forall u, v \in H^1(\Omega)$$

and the induced norm $||\cdot||_{H^2(\Omega)}$.

Remark. *In view of these definitions, we can see that*

$$||u||^2_{H^1(\Omega)} = ||u||^2_{L^2(\Omega)} + |u|^2_{H^1\Omega)}$$

with $|u|_{H^1\Omega)} = ||\nabla u||_{L^2(\Omega)}$ called H^1 semi-norm.
We also note that

$$||u||^2_{H^2(\Omega)} = ||u||^2_{L^2(\Omega)} + |u|^2_{H^1(\Omega)} + |u|^2_{H^2(\Omega)}$$

with $|u|_{H^2\Omega)} = ||u''||_{L^2(\Omega)}$ called H^2 semi-norm.

Remark. *In the following, we will note $||\cdot||_{0,\Omega}$ the L^2 norm on Ω, $||\cdot||_{1,\Omega}$ the H^1 norm on Ω and $||\cdot||_{2,\Omega}$ the H^2 norm on Ω. We will also note $|\cdot|_{1,\Omega}$ the H^1 semi-norm on Ω and $|\cdot|_{2,\Omega}$ the H^2 semi-norm on Ω.*

2.1.2 General principle of the method

Let's consider a domain Ω whose boundary is denoted $\partial\Omega$. We seek to determine a function u defined on Ω, solution of a partial differential equation (PDE) for given boundary conditions. The general approach of the finite element method is to write down the variational formulation of this PDE, thus giving us a problem of the following type:
Variational Problem :

$$\text{Find } u \in V \text{ such that } a(u, v) = l(v),\ \forall v \in V$$

where V is a Hilbert space, a is a bilinear form and l is a linear form.
To do this, we multiply the PDE by a test function $v \in V$, then integrate over $L^2(\Omega)$.
The idea of FEM is to use Galerkin's method. We then look for an approximate solution u_h in V_h, a finite-dimensional space dependent on a positive parameter h such that

$$V_h \subset V, \quad \dim V_h = N_h < \infty, \quad \forall h > 0$$

The variational problem can then be approached by :
Approach Problem :

$$\text{Find } u_h \in V_h \text{ such that } a(u_h, v_h) = l(v_h),\ \forall v_h \in V$$

As V_h is of finite dimension, we can consider a basis $(\varphi_1, \ldots, \varphi_{N_h})$ of V_h and thus decompose u_h on this basis as :

$$u_h = \sum_{i=1}^{N_h} u_i \varphi_i \tag{1}$$

The approached problem is then rewritten as

$$\text{Find } u_1, \ldots, u_{N_h} \text{ such that } \sum_{i=1}^{N_h} u_i\, a(\varphi_i, v_h) = l(v_h),\ \forall v_h \in V$$

and

$$\text{Find } u_1, \ldots, u_{N_h} \text{ such that } \sum_{i=1}^{N_h} u_i\, a(\varphi_i, \varphi_j) = l(\varphi_j),\ \forall j \in \{1, \ldots, N_h\}$$

Solving the PDE involves solving the following linear system:

$$AU = b$$

with

$$A = (a(\varphi_i, \varphi_j))_{1 \le i,j \le N_h}, \quad U = (u_i)_{1 \le i \le N_h} \quad \text{and} \quad b = (l(\varphi_j))_{1 \le j \le N_h}$$

Remark. *To impose Dirichlet boundary conditions, we can use one of 2 methods. The elimination method consists in modifying the rows associated with the boundary nodes in the finite element matrix. More precisely, we set the rows to 0 except 1 on the diagonal and the value of the condition on the second member. In other words, we simply write the value of the degrees of freedom at the Dirichlet boundary. The penalization method consists in modifying the matrix and the second member as follows:*

$$A_{i,i} := A_{i,i} + \frac{1}{\epsilon}$$
$$f_i := f_i + \frac{1}{\epsilon} g_i$$

with $\epsilon > 0$ *and* i *is a boundary nodes.*

2.1.3 Some details on FEM

After having seen the general principle of FEM, it remains to define the V_h spaces and the $\{\varphi_i\}$ basis functions.

Remark. *The choice of V_h space is fundamental to have an efficient method that gives a good approximation u_h of u. In particular, the choice of the $\{\varphi_i\}$ basis of V_h influences the structure of the A matrix in terms of its sparsity and its condition number.*

To do this, we'll need several notions, which will be detailed in the following sections. First, we'll need to generate a **mesh** of our Ω domain. This will enable us to solve the PDE discretely at selected points. This is where the notion of **finite Lagrange elements** comes in. The properties of these elements, particularly in terms of their **affine family of finite elements**, is a key point of the method, which will enable us to bring each element of the mesh back to a **reference element** by using a **geometric transformation**. To describe these steps, we'll need to know 2 basic concepts: the **unisolvence** principle and the definitions of the **polynomial spaces** used ($\mathbb{P}_k$ and $\mathbb{Q}_k$).

2.1.3.1 Unisolvance

Definition 2.3. *Let $\Sigma = \{a_1, \dots, a_N\}$ be a set of N distinct points of $\mathbb{R}^n$. Let P be a finite-dimensional vector space of $\mathbb{R}^n$ functions taking values in $\mathbb{R}$. We say that Σ is P-unisolvent if and only if for all real $\alpha_1, \dots, \alpha_N$, there exists a unique element p of P such that $p(a_i) = \alpha_i, i = 1, \dots, N$. This means that the function*

$$\begin{aligned} L : P &\to \mathbb{R}^N \\ p &\mapsto (p(a_1), \dots, p(a_N)) \end{aligned}$$

is bijective.

Remark. *In practice, to show that Σ is P-unisolvent, we simply check that $\dim P = card\,(\Sigma)$ and then prove the injectivity or surjectivity of L. The injectivity of L is demonstrated by showing that the only function of P that annuls on all points of Σ is the null function. The surjectivity of L is shown by identifying a family $p_1, \dots, p_N$ of elements of P such that $p_i(a_j) = \delta_{ij}$. Given real $\alpha_1, \dots, \alpha_N$, the function $p = \sum_{i=1}^{N} \alpha_i p_i$ then verifies $p(a_j) = \alpha_j, j = 1 \dots, N$.*

Remark. *We call local basis functions of element K the N functions $p_1, \dots, p_N$ of P such that*

$$p_i(a_j) = \delta_{ij}, \quad 1 \leq i, j \leq N$$

2.1.3.2 Polynomial space

Let $\mathbb{P}_k$ be the vector space of polynomials of total degree less than or equal to k.

- In $\mathbb{R}$: $\mathbb{P}_k = \text{Vect}\{1, X, \dots, X^k\}$ and $\dim \mathbb{P}_k = k + 1$

- In $\mathbb{R}^2$: $\mathbb{P}_k = \text{Vect}\{X^i Y^j, 0 \leq i+j \leq k\}$ and $\dim \mathbb{P}_k = \frac{(k+1)(k+2)}{2}$
- In $\mathbb{R}^3$: $\mathbb{P}_k = \text{Vect}\{1, X^i Y^j Z^l, 0 \leq i+j+l \leq k\}$ and $\dim \mathbb{P}_k = \frac{(k+1)(k+2)(k+3)}{6}$

Let $\mathbb{Q}_k$ be the vector space of polynomials of degree less than or equal to k with respect to each variable.

- In $\mathbb{R}$: $\mathbb{Q}_k = \mathbb{P}_k$.
- In $\mathbb{R}^2$: $\mathbb{Q}_k = \text{Vect}\{X^i Y^j, 0 \leq i, j \leq k\}$ and $\dim \mathbb{Q}_k = (k+1)^2$
- In $\mathbb{R}^3$: $\mathbb{Q}_k = \text{Vect}\{1, X^i Y^j Z^l, 0 \leq i, j, l \leq k\}$ and $\dim \mathbb{Q}_k = (k+1)^3$

Remark. *In practice, we will use the $\mathbb{P}^k$ family for triangles/tetrahedra and $\mathbb{Q}_k$ for quadrilaterals.*

2.1.3.3 Finite Lagrange Element

The most classic and simplest type of finite element is the Lagrange finite element.

Definition 2.4 (Lagrange Finite Element)**.** *A finite Lagrange element is a triplet (K, Σ, P) such that*

- *K is a geometric element of $\mathbb{R}^n$ ($n = 1, 2$ or 3), compact, connected and of non-empty interior.*
- *$\Sigma = \{a_1, \ldots, a_N\}$ is a finite set of N distinct points of K.*
- *P is a finite-dimensional vector space of real functions defined on K and such that Σ is P-unisolvent (so $\dim P = N$).*

Example. *Let K be the segment $[a_1, a_2]$. Let's show that $\Sigma = \{a_1, a_2\}$ is P-unisolvent for $P = \mathbb{P}^1$.*
Since $\{1, x\}$ is a base of $\mathbb{P}^1$, we have $\dim P = \text{card}\, \Sigma = 2$.
Moreover, we can write $p_i = \alpha_i x + \beta_i, i = 1, 2$. Thus

$$\begin{cases} p_1(a_1) = 1 \\ p_1(a_2) = 0 \end{cases} \iff \begin{cases} \alpha_1 a_1 + \beta_1 = 1 \\ \alpha_1 a_2 + \beta_1 = 0 \end{cases} \iff \begin{cases} \alpha_1 = \dfrac{1}{a_1 - a_2} \\ \beta_1 = -\dfrac{a_2}{a_1 - a_2} \end{cases}$$

and

$$\begin{cases} p_2(a_1) = 0 \\ p_2(a_2) = 1 \end{cases} \iff \begin{cases} \alpha_2 a_1 + \beta_2 = 0 \\ \alpha_2 a_2 + \beta_2 = 1 \end{cases} \iff \begin{cases} \alpha_1 = \dfrac{1}{a_2 - a_1} \\ \beta_1 = -\dfrac{a_1}{a_2 - a_1} \end{cases}$$

Thus

$$p_1(x) = \frac{x - a_2}{a_1 - a_2} \quad \textit{and} \quad p_2(x) = \frac{x - a_1}{a_2 - a_1}$$

We deduce the surjectivity of L and Σ is $\mathbb{P}^1$-unisolvent.
Thus (K, Σ, P) is a Lagrange Finite Element.

Definition 2.5. *Two finite elements* $(\hat{K}, \hat{\Sigma}, \hat{P})$ *and* (K, Σ, P) *are affine-equivalent if and only if there exists an irreversible affine function* F *such that*

- $K = F(\hat{K})$
- $a_i = F(\hat{a}_i), i = 1, \ldots, N$
- $P = \{\hat{p} \circ F^{-1}, \hat{p} \in \hat{P}\}$.

We then call an ***affine family of finite elements*** *a family of finite elements, all affine-equivalent to the same element* $(\hat{K}, \hat{\Sigma}, \hat{P})$*, called the* ***reference element****.*

Remark. *Let* $(\hat{K}, \hat{\Sigma}, \hat{P})$ *and* (K, Σ, P) *be two affine-equivalent finite elements, via an* F *transformation. Let* $\hat{p}_i$ *be the local basis functions on* $\hat{K}$*. Then the local basis functions on* K *are* $p_i = \hat{p}_i \circ F^{-1}$*.*

Remark. *In practice, working with an affine family of finite elements means that all integral calculations can be reduced to calculations on the reference element.*
The reference elements in 1D, 2D triangular and 3D tetrahedral are :

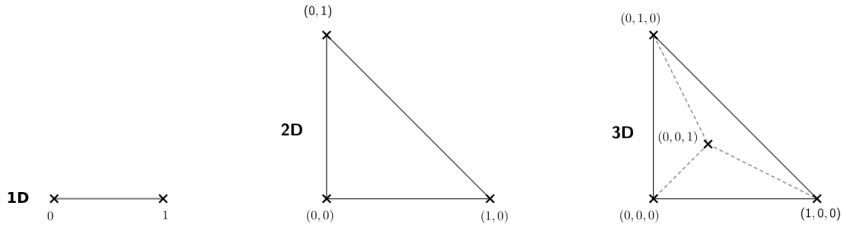

Figure 2.1: Example of reference Elements.

2.1.3.4 Mesh

In 1D, the construction of a mesh consists in creating a subdivision of the interval $[a, b]$. We can extend this definition in 2D and 3D by considering that a mesh is formed by a family of elements $\mathcal{T}_h = \{K_1, \ldots, K_{N_e}\}$ (see Fig 2.2) where N_e is the number of elements.
In 2D, these elements can be triangles or rectangles. In 3D, they can be tetrahedrons, parallelepipeds or prisms.

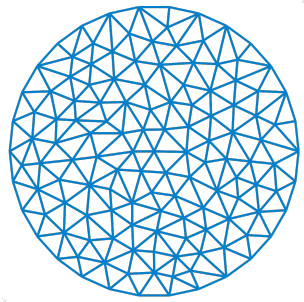

Figure 2.2: Example of a triangular mesh on a circles.

2.1.3.5 Construction of V_h space

Geometric transformation : A mesh is generated by

- A reference element noted $\hat{K}$.
- A family of geometric transformations mapping $\hat{K}$ to the elements $K_1, \ldots, K_{N_e}$. Thus, for a cell $K \in \mathcal{T}_h$, we denote T_K the geometric transformation mapping $\hat{K}$ to K :

$$T_K : \hat{K} \to K$$

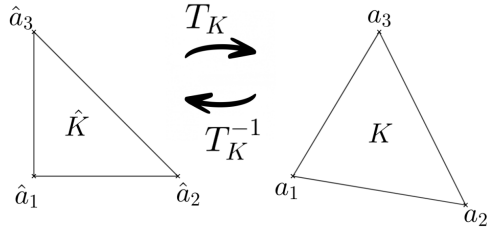

Figure 2.3: Geometric transformation applied to a triangle.

Let $(\hat{K}, \hat{\Sigma}, \hat{P})$ be the finite reference element with

- the degrees of freedom of the reference element $\hat{K}$: $\hat{\Sigma} = \{\hat{a}_1, \ldots, \hat{a}_{n_f}\}$ with n_f the number of degrees of freedom.
- the local basis functions of $\hat{K}$: $\{\hat{\psi}_1, \ldots, \hat{\psi}_{n_f}\}$ (also called form functions)

So for each $K \in \mathcal{T}_h$, we consider a tuple $\{a_{K,1}, \ldots, a_{K,n_f}\}$ (degrees of freedom) and the associated geometric transformation is defined by :

$$T_K : \hat{x} \mapsto \sum_{i=1}^{n_f} a_{K,i} \hat{\psi}_i(\hat{x})$$

In particular, we have

$$T_K(\hat{a}_i) = a_{K,i}, \quad i = 1, \ldots, n_f$$

Remark. *In particular, if the form functions are affine, the geometric transformations will be too. This is an interesting property, as the gradient of these geometric transformations will be constant.*

Remark. *In the following, we will assume that these transformations are C^1-diffeomorphisms (i.e. the transformation and its inverse are C^1 and bijective).*

Construction of the basis (φ_i) of V_h :
For each $K \in \mathcal{T}_h$, let (K, Σ, P) be an finite element with

- the degrees of freedom of the element K : $\Sigma = \{a_{K,i} = T_K(\hat{a}_i), i = 1, \dots, n_f\}$
- the local basis functions of K: $\{\psi_{K,i} = \hat{\psi}_i \circ T_K^{-1}, i = 1, \dots, n_f\}$ (because $(\hat{K}, \hat{\Sigma}, \hat{P})$ and (K, Σ, P) are affine-equivalent).

By noting $\{a_1, \dots, a_{N_f}\} = \bigcup_{K \in \mathcal{T}_h} \{a_{K,1}, \dots, a_{K,n_f}\}$ with N_f the total number of degrees of freedom (over all the geometry), we have

$$\forall j \in \{1, \dots, N_f\}, \quad \varphi_{j|K} = \psi_{K,a_{K,j}}$$

The ϕ_j functions are then in the space of piece-wise affine continuous functions, defined by

$$P^k_{C,h} = \{v_h \in C^0(\bar{\Omega}), \forall K \in \mathcal{T}_h, v_{h|K} \in \mathbb{P}_k\} \subset H^1(\Omega)$$

In fact, the functions $\{\varphi_1, \dots, \varphi_{N_f}\}$ form a basis of $P^k_{C,h}$ and so we can choose $V_h = P^k_{C,h}$.

2.1.4 Application to the Poisson problem

Weak formulation :
We want to apply the standard FEM method to the Poisson problem with Dirichlet condition under consideration. Let's start by writing the variational formulation of the problem. We start with the strong formulation of the problem :

$$-\Delta u = f \text{ on } \Omega$$

Multiplying by a test function $v \in H^1_0(\Omega)$ and integrating over Ω, we obtain

$$-\int_\Omega \Delta u v = \int_\Omega f v.$$

By integration by parts, we have

$$-\int_\Omega \Delta u v = \int_\Omega \nabla u \cdot \nabla v - \int_\Gamma \frac{\partial u}{\partial n} v.$$

This leads to the following weak formulation

$$\text{Find } u \in V \text{ such that } a(u, v) = l(v), \ \forall v \in V$$

with

$$\begin{cases} a(u, v) = \displaystyle\int_\Omega \nabla u \cdot \nabla v \\ l(v) = \displaystyle\int_\Omega f v \end{cases}$$

because $v \in H^1_0(\Omega)$.
Unicity of the solution :
An important result of the FEM method is the following theorem, which shows the unicity of the solution:

Proposition 2.1 (Lax-Milgram)**.** *Let a be a continuous, coercive bilinear form on V and l a continuous, linear form on V. Then the variational problem has a unique solution $u \in V$. Moreover, if the bilinear form is symmetrical, u is a solution to the following minimization problem:*

$$J(u) = \min_{v \in V} J(v), \quad J(v) = \frac{1}{2} a(v,v) - l(v)$$

It can then be shown that the Poisson problem with Dirichlet condition has a unique weak solution $u \in H^2(\Omega)$.

2.2 ϕ-FEM

In this section, we will present the ϕ-FEM method. We will first present the context in which the method is used and its general principle (Section 2.2.1. Next, we will give a general presentation of the method, starting with a description of the spaces required (Section 2.2.2.1), followed by a description of the ϕ-FEM method (Section 2.2.2.2). Finally, we will give some details on the ϕ-FEM method (section 2.2.3).

2.2.1 Context and general principle of the method

The ϕ-FEM method is a new fictitious domain finite element method that does not require a mesh conforming to the real boundary. In the context of augmented surgery, this method presents a considerable advantage. During real-time simulation, the geometry (in our specific context, an organ such as the liver, for example) can deform over time. Methods such as standard FEM, which requires a mesh fitted to the boundary, necessitate a complete remeshing of the geometry at each time step (Figure 2.4). Unlike this type of method, ϕ-FEM requires only the generation of a single mesh : the mesh of a fictitious domain containing the entire geometry (Figure 2.5). As the boundary of the geometry is represented by a level-set function ϕ, only this function will change over time, which is a real time-saver.

Remark. *Note that changing the ϕ function creates new sets of cells, all of them described in the Section 2.2.2.1.*

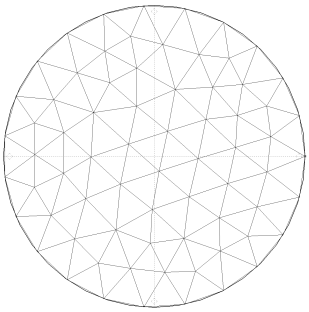

Figure 2.4: Standard FEM mesh example.

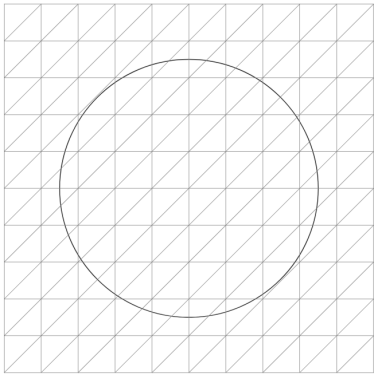

Figure 2.5: ϕ-FEM mesh example.

Remark. *For the purposes of this internship, the geometries considered are not organs (such as the liver), because these are complex geometries. We are considering simpler geometries such as circles or squares. It is also important to note that the ϕ-FEM method has a considerable advantage: by constructing a fictitious mesh around the domain, we can generate a Cartesian mesh. This type of mesh can easily be represented by matrices, in the same way as images, hence the possibility of teaching these ϕ-FEM solutions to an FNO who generally works on images. A paper in progress presents results with the combination of PhiFEM and an FNO on more complex geometries, notably ellipses.*

2.2.2 General presentation of the ϕ-FEM method

In this section, we consider the case of the Poisson problem with homogeneous Dirichlet condition ($g = 0$ on Γ). For the case of non-homogeneous Dirichlet conditions, we will give more details in Section 2.2.3.2. For more details on mesh assumptions, convergence results and finite element matrix condition number, please refer to [9]. ϕ-FEM schemes for the Poisson problem with Neumann or mixed (Dirichlet and Neumann) conditions are presented in [8, 6]. The ϕ-FEM scheme can also be found for other PDEs, including linear elasticity [6, Chapter 2], the heat equation [6, Chapter 5] and the Stokes problem [7].

2.2.2.1 Description of spaces

As previously said, we will consider the Poisson-Dirichlet problem

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = g, & \text{on } \partial\Omega, \end{cases} \tag{2}$$

where the domain Ω and its boundary Γ are given by a level-set function ϕ such that

$$\Omega = \{\phi < 0\} \quad \text{and} \quad \Gamma = \{\phi = 0\}.$$

The level-set function ϕ is supposed to be known on $\mathbb{R}^d$, sufficiently smooth, and to behave near Γ as the signed distance to Γ (Figure 2.6).

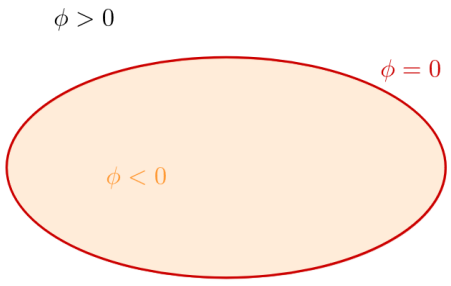

Figure 2.6: Definition of the level-set function.

Example. *If Ω is a circle of center A of coordinates (x_A, y_A) and radius r, the level-set function can be defined by*

$$\phi(x,y) = -r^2 + (x-x_A)^2 + (y-y_A)^2.$$

If Ω is an ellipse with center A of coordinates (x_A, y_A) and parameters (a,b), the level-set function can be defined by

$$\phi(x,y) = -1 + \frac{(x-x_A)^2}{a^2} + \frac{(y-y_A)^2}{b^2}.$$

We assume that Ω is inside a domain $\mathcal{O}$ and we introduce a simple quasi-uniform mesh $\mathcal{T}_h^{\mathcal{O}}$ on $\mathcal{O}$ (Figure 2.7).
We introduce now an approximation $\phi_h \in V_{h,\mathcal{O}}^{(l)}$ of ϕ given by $\phi_h = I_{h,\mathcal{O}}^{(l)}(\phi)$ where $I_{h,\mathcal{O}}^{(l)}$ is the standard Lagrange interpolation operator on

$$V_{h,\mathcal{O}}^{(l)} = \left\{ v_h \in H^1(\mathcal{O}) : v_{h|_T} \in \mathbb{P}_l(T) \; \forall T \in \mathcal{T}_h^{\mathcal{O}} \right\}$$

and we denote by $\Gamma_h = \{\phi_h = 0\}$, the approximate boundary of Γ (Figure 2.8).
We will consider $\mathcal{T}_h$ a sub-mesh of $\mathcal{T}_h^{\mathcal{O}}$ obtained by removing the elements located entirely outside Ω (Figure 2.8). To be more specific, $\mathcal{T}_h$ is defined by

$$\mathcal{T}_h = \left\{ T \in \mathcal{T}_h^{\mathcal{O}} : T \cap \{\phi_h < 0\} \neq \emptyset \right\}.$$

We denote Ω_h the domain covered by the $\mathcal{T}_h$ mesh (Ω_h will be slightly larger than Ω) and $\partial\Omega_h$ its boundary (Figure 2.8). The domain Ω_h is defined by

$$\Omega_h = \left(\cup_{T \in \mathcal{T}_h} T \right)^O.$$

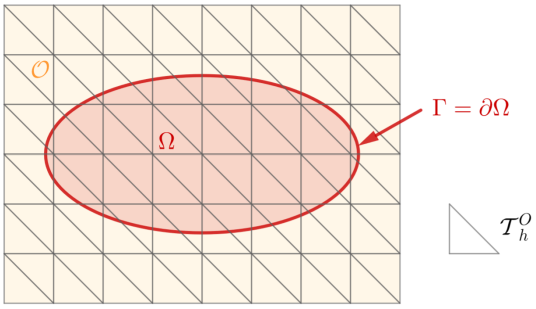

Figure 2.7: Fictitious domain.

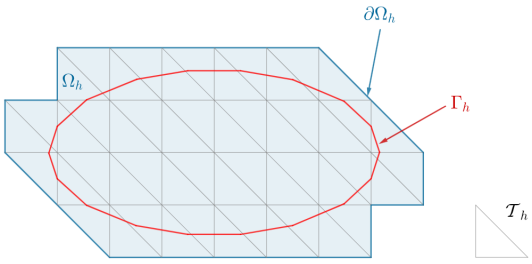

Figure 2.8: Domain considered.

Now, we can introduce $\mathcal{T}_h^{\Gamma} \subset \mathcal{T}_h$ (Figure 2.9) which contains the mesh elements cut by the approximate boundary $\Gamma_h = \{\phi_h = 0\}$, i.e.

$$\mathcal{T}_h^{\Gamma} = \{ T \in \mathcal{T}_h : T \cap \Gamma_h \neq \emptyset \},$$

and $\mathcal{F}_h^{\Gamma}$ (Figure 2.10) which collects the interior facets of the mesh $\mathcal{T}_h$ either cut by Γ_h or belonging to a cut mesh element

$$\mathcal{F}_h^{\Gamma} = \{ E \text{ (an internal facet of } \mathcal{T}_h) \text{ such that } \exists T \in \mathcal{T}_h : T \cap \Gamma_h \neq \emptyset \text{ and } E \in \partial T \}.$$

We denote by Ω_h^Γ the domain covered by the $\mathcal{T}_h^\Gamma$ mesh (Figure 2.9) and also defined by

$$\Omega_h^\Gamma = \left(\cup_{T\in\mathcal{T}_h^\Gamma} T\right)^O.$$

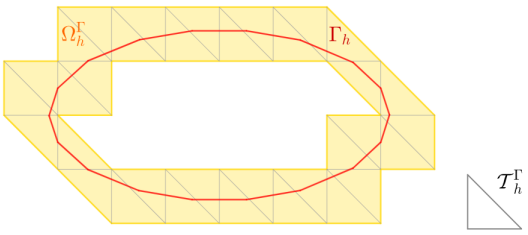

Figure 2.9: Boundary cells.

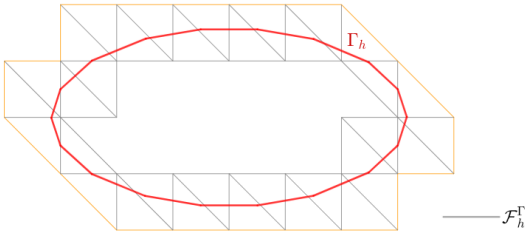

Figure 2.10: Boundary edges.

2.2.2.2 Description of the ϕ-FEM method

As with standard FEM, the general idea behind ϕ-FEM is to find a weak solution (i.e. a solution to the variational problem) to the considered problem (2). The main difference lies in the spaces considered. In fact, we are no longer looking to solve the problem on Ω (of boundary Γ) but on Ω_h (of boundary $\partial\Omega_h$). Since our boundary conditions are defined on Γ, we don't have a direct condition on the $\partial\Omega_h$ boundary, so we will have to add terms to the variational formulation of the problem, called stabilization terms.
Assuming that the right-hand side f is currently well-defined on Ω_h and that the solution u can be extended on Ω_h such that $-\Delta u = f$ on Ω_h, we can introduce a new unknown $w \in H^1(\Omega_h)$ such that $u = \phi w$ and the boundary condition on Γ is satisfied (since $\phi = 0$ on Γ). After an integration by parts, we have

$$\int_{\Omega_h} \nabla(\phi w)\cdot\nabla(\phi v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi w)\phi v = \int_{\Omega_h} f\phi v, \quad \forall v \in H^1(\Omega_h).$$

Remark. *Note that Ω_h is constructed using ϕ_h and therefore implicitly depends on ϕ.*

Given an approximation ϕ_h of ϕ on the mesh $\mathcal{T}_h$, as defined in Section 2.2.2.1, and a finite element space V_h on $\mathcal{T}_h$, we can then search for $w_h \in V_h$ such that

$$a_h(w_h, v_h) = l_h(v_h), \quad \forall v_h \in V_h.$$

We can consider the finite element space $V_h = V_h^{(k)}$ with

$$V_h^{(k)} = \left\{v_h \in H^1(\Omega_h) : v_{h|_T} \in \mathbb{P}_k(T)\ \forall T \in \mathcal{T}_h\right\}.$$

The bilinear form a_h and the linear form l_h are defined by

$$a_h(w, v) = \int_{\Omega_h} \nabla(\phi_h w)\cdot\nabla(\phi_h v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w)\phi_h v + G_h(w, v)$$

and

$$l_h(v) = \int_{\Omega_h} f\phi_h v + G_h^{rhs}(v)$$

with

$$G_h(w,v) = \sigma h \sum_{E\in\mathscr{F}_h^\Gamma}\int_E \left[\frac{\partial}{\partial n}(\phi_h w)\right]\left[\frac{\partial}{\partial n}(\phi_h v)\right] + \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T \Delta(\phi_h w)\Delta(\phi_h v)$$

and

$$G_h^{rhs}(v) = -\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T f\Delta(\phi_h v).$$

with σ an independent parameter of h, which we'll call the stabilization parameter.

Remark. *Note that* $[\cdot]$ *is the jump on the interface E defined by*

$$\left[\frac{\partial}{\partial n}(\phi_h w)\right] = \nabla(\phi_h w)^+ \cdot n - \nabla(\phi_h w)^- \cdot n$$

with n is the unit normal vector outside E.

2.2.3 Some details on ϕ-FEM

In this section, we first give some information on stabilization terms (Section 2.2.3.1) and then present two methods for imposing non-homogeneous Dirichlet conditions, the direct method and the dual method (Section 2.2.3.2).

2.2.3.1 Stabilization terms

As introduced previously, the stabilization terms are intended to reduce the errors created by the "fictitious" boundary, but they also have the effect of ensuring the correct condition number of the finite element matrix and permitting to restore the coercivity of the bilinear scheme.
The first term of $G_h(w,v)$ defined by

$$\sigma h \sum_{E\in\mathscr{F}_h^\Gamma}\int_E \left[\frac{\partial}{\partial n}(\phi_h w)\right]\left[\frac{\partial}{\partial n}(\phi_h v)\right]$$

is a first-order stabilization term. This stabilization term is based on [4]. It also ensures the continuity of the solution by penalizing gradient jumps.
By subtracting $G_h^{rhs}(v)$ from the second term of $G_h(w,v)$, i.e.

$$\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T \Delta(\phi_h w)\Delta(\phi_h v) + \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T f\Delta(\phi_h v),$$

which can be rewritten as

$$\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T \big(\Delta(\phi_h w) + f\big)\Delta(\phi_h v),$$

we recognize the strong formulation of the Poisson problem. This second-order stabilization term penalizes the scheme by requiring the solution to verify the strong form on Ω_h^Γ. In fact, this term cancels out if $\phi_h w$ is the exact solution of the Poisson problem under consideration.

2.2.3.2 Non-homogeneous case

In the case of a non-homogeneous Dirichlet condition, we want to impose $u = g$ on Γ. To do this, we will consider 2 approaches introduced in [6] and presented below:

- **Direct method :** In this method, we must suppose that g is currently given over the entire Ω_h and not just over Γ. We can then write the solution u as

$$u = \phi w + g, \text{ on } \Omega_h.$$

  It can then be injected into the weak formulation of the homogeneous problem and we can then search for w_h on Ω_h such that

$$\int_{\Omega_h} \nabla(\phi_h w_h)\nabla(\phi_h v_h) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi_h w_h)\phi_h v_h + G_h(w_h, v_h) = \int_{\Omega_h} f\phi_h v_h \\ - \int_{\Omega_h} \nabla g \nabla(\phi_h v_h) + \int_{\partial\Omega_h} \frac{\partial g}{\partial n}\phi_h v_h + G_h^{rhs}(v_h),\ \forall v_h \in \Omega_h$$

  with

$$G_h(w, v) = \sigma h \sum_{E\in\mathscr{F}_h^\Gamma} \int_E \left[\frac{\partial}{\partial n}(\phi_h w)\right]\left[\frac{\partial}{\partial n}(\phi_h v)\right] + \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma} \int_T \Delta(\phi_h w)\Delta(\phi_h v)$$

  and

$$G_h^{rhs}(v) = -\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma} \int_T f\Delta(\phi_h v) - \sigma h \sum_{E\in\mathscr{F}_h^\Gamma} \int_E \left[\frac{\partial g}{\partial n}\right]\left[\frac{\partial}{\partial n}(\phi_h v)\right] - \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma} \int_T \Delta g \Delta(\phi_h v)$$

- **Dual method :** We now assume that g is defined on Ω_h^Γ and not on Ω_h. We then introduce a new unknown p on Ω_h^Γ in addition to the unknown u on Ω_h and so we aim to impose

$$u = \phi p + g, \text{ on } \Omega_h^\Gamma.$$

  So we look for u on Ω_h and p on Ω_h^Γ such that

$$\int_{\Omega_h} \nabla u \nabla v - \int_{\partial\Omega_h} \frac{\partial u}{\partial n} v + \frac{\gamma}{h^2} \sum_{T\in\mathscr{T}_h^\Gamma} \int_T \left(u - \frac{1}{h}\phi p\right)\left(v - \frac{1}{h}\phi q\right) + G_h(u, v) = \int_{\Omega_h} f v \\ + \frac{\gamma}{h^2} \sum_{T\in\mathscr{T}_h^\Gamma} \int_T g\left(v - \frac{1}{h}\phi q\right) + G_h^{rhs}(v),\ \forall v \text{ on } \Omega_h,\ q \text{ on } \Omega_h^\Gamma.$$

with γ an other positive stabilization parameter,

$$G_h(u,v) = \sigma h \sum_{E \in \mathscr{F}_h^\Gamma} \int_E \left[\frac{\partial u}{\partial n}\right]\left[\frac{\partial v}{\partial n}\right] + \sigma h^2 \sum_{T \in \mathscr{T}_h^\Gamma} \int_T \Delta u \Delta v$$

and

$$G_h^{rhs}(v) = -\sigma h^2 \sum_{T \in \mathscr{T}_h^\Gamma} \int_T f \Delta v.$$

Remark. *The factors $\frac{1}{h}$ and $\frac{1}{h^2}$ control the condition number of the finite element matrix. For more details, please refer to the article [8].*

Remark. *In the context of this internship, we won't be concerned with the choice of stabilization parameters σ and γ. We'll always take $\sigma = 20$ and $\gamma = 1$, but it's important to note that they can have a significant influence on the results.*

3 Fourier Neural Operator (FNO)

We will now introduce Fourier Neural Operators (FNO), which belong to the category of so-called neural operator networks. Unlike standard neural networks, which learn using inputs and outputs of fixed dimensions, neural operators learn operators, which are mappings between spaces of functions. They can be evaluated at any data resolution without the need for retraining. As a result, they are widely used in PDE solving and constitute an active field of research. For more information, please refer to the following article [11, 12, 10, 13].
In image treatment, we call image tensors of size $ni \times nj \times nk$, where $ni \times nj$ corresponds to the image resolution and nk corresponds to its number of channels. For example, an RGB (Red Green Blue) image has $nk = 3$ channels. We choose here to present the FNO as an operator acting on discrete images. The reference article [11] present it in its continuous aspect, which is an interesting point of view. Indeed, it is thanks to this property that it can be trained/evaluated with images of different resolutions.

Remark. *The FNO used was implemented by Vincent Vigon[6] using Python's tensorflow library[7]. Furthermore, note that this report does not include a test of model parameter variation.*

3.1 Architecture of the FNO

The following figure (Figure 3.1) describes the FNO architecture in detail:

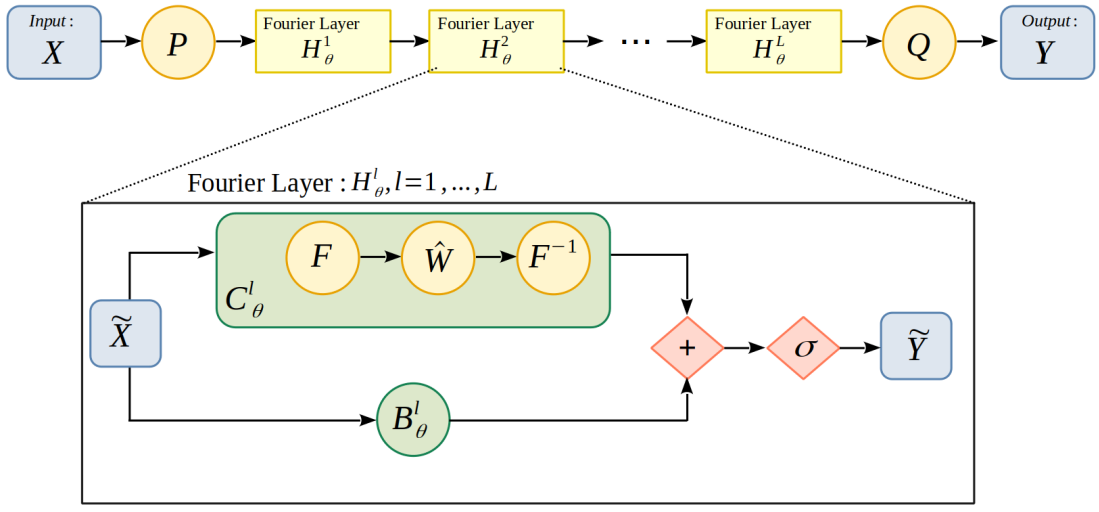

Figure 3.1: Architecture of the FNO.

The architecture of the FNO is as follows:

$$G_\theta = Q \circ \mathcal{H}^L_\theta \circ \cdots \circ \mathcal{H}^1_\theta \circ P$$

[6] Vincent Vigon: https://irma.math.unistra.fr/ vigon/
[7] Tensorflow: https://www.tensorflow.org/?hl=fr

We'll now describe the composition of the Figure 3.1 in a little more detail :

- We start with input X of shape (batch_size, height, width, nb_channels) with batch_size the number of images to be processed at the same time, height and width the dimensions of the images and nb_channels the number of channels. Simplify by (bs,ni,nj,nk).
- We perform a P transformation in order to move to a space with more channels. This step enables the network to build a sufficiently rich representation of the data. For example, a Dense layer (also known as fully-connected) can be used.
- We then apply L Fourier layers, noted $\mathscr{H}_\theta^l$, $l = 1,\dots,L$, whose specifications will be detailed in Section 3.2.
- We then return to the target dimension by performing a Q transformation. In our case, the number of output channels is 1.
- We then obtain the output of the Y model of shape (bs,ni,nj,1).

Remark. *Note that the P and Q layers are in fact fully-connected multi-layer perceptrons, which means that they perform local transformations at each point, and therefore do not depend on the mesh resolution considered.*
Fourier layers are also independent of mesh resolution. Indeed, as we learn in Fourier space, the value of the Fourier modes does not depend on the mesh resolution.
We deduce that the entire FNO does not depend on the mesh resolution.

3.2 Fourier Layer structure

Each Fourier layer is divided into two sub-layers:

$$\tilde{Y} = \mathscr{H}_\theta^l(\tilde{X}) = \sigma\left(\mathscr{C}_\theta^l(\tilde{X}) + \mathscr{B}_\theta^l(\tilde{X})\right)$$

where

- $\tilde{X}$ corresponds to the input of the current layer and $\tilde{Y}$ to the output.
- σ is an activation function. For $l = 1,\dots,L-1$, we'll take the activation function ReLU (Rectified Linear Unit) and for $l = L$ we'll take the activation function GELU (Gaussian Error Linear Units).

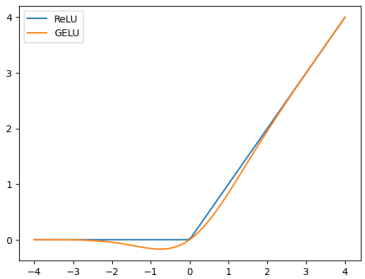

Figure 3.2: Activation functions used.

- $\mathscr{C}_\theta^l$ is a convolution layer where convolution is performed by FFT (Fast Fourier Transform). For more details, see Section 3.2.1.
- $\mathscr{B}_\theta^l$ is the "bias-layer". For more details, see Section 3.2.2.

3.2.1 Convolution sublayer

Each $\mathscr{C}_\theta^l$ convolution layer contains a trainable kernel $\hat{W}$ and performs the transformation

$$\mathscr{C}_\theta^l(X) = \mathscr{F}^{-1}(\mathscr{F}(X) \cdot \hat{W})$$

where $\mathscr{F}$ corresponds to the 2D Discrete Fourier Transform (DFT) on a $ni \times nj$ resolution grid and

$$(Y \cdot \hat{W})_{ijk} = \sum_{k'} Y_{ijk'} \hat{W}_{ijk'}$$

In other words, this transformation is applied channel by channel.

Remark. *An image is fundamentally a signal. Just as 1D signals show changes in amplitude (sound) over time, 2D signals show variations in intensity (light) over space. The Fourier transform allows us to move from the spatial or temporal domain into the frequency domain. In a sound signal (1D signal), low frequencies represent low-pitched sounds and high frequencies represent high-pitched sounds. In the case of an image (2D signal), low frequencies represent large homogeneous surfaces and blurred parts, while high frequencies represent contours, more generally abrupt changes in intensity and, finally, noise.*

The 2D DFT is defined by :

$$\mathscr{F}(X)_{ijk} = \frac{1}{ni}\frac{1}{nj}\sum_{i'=0}^{ni-1}\sum_{j'=0}^{nj-1} X_{i'j'k} e^{-2\sqrt{-1}\pi\left(\frac{ii'}{ni}+\frac{jj'}{nj}\right)}$$

The inverse of the 2D DFT is defined by :

$$\mathscr{F}^{-1}(X)_{ijk} = \sum_{i'=0}^{ni-1}\sum_{j'=0}^{nj-1} X_{i'j'k} e^{2\sqrt{-1}\pi\left(\frac{ii'}{ni}+\frac{jj'}{nj}\right)}$$

We can easily show that $\mathscr{F}$ is the reciprocal function of $\mathscr{F}^{-1}$. We have

$$\begin{aligned}
\mathscr{F}^{-1}(\mathscr{F}(X))_{ijk} &= \sum_{i'=0}^{ni-1}\sum_{j'=0}^{nj-1} \mathscr{F}(X)_{i'j'k} e^{2\sqrt{-1}\pi\left(\frac{ii'}{ni}+\frac{jj'}{nj}\right)} \\
&= \frac{1}{ni}\frac{1}{nj}\sum_{i'j'}\sum_{i''j''} X_{i''j''k} e^{-2\sqrt{-1}\pi\left(\frac{i'i''}{ni}+\frac{j'j''}{nj}\right)} e^{2\sqrt{-1}\pi\left(\frac{ii'}{ni}+\frac{jj'}{nj}\right)} \\
&= \frac{1}{ni}\frac{1}{nj}\sum_{i''j''} X_{i''j''k} \sum_{i'j'} e^{2\sqrt{-1}\pi\frac{i'}{ni}(i-i'')} e^{2\sqrt{-1}\pi\frac{j'}{nj}(j-j'')}
\end{aligned}$$

Let

$$S = \sum_{i'j'} e^{2\sqrt{-1}\pi \frac{i'}{ni}(i-i'')} e^{2\sqrt{-1}\pi \frac{j'}{nj}(j-j'')}$$

Thus

- If $(i,j) = (i'',j'') : S = \sum_{i',j'} 1 = ni \times nj$
- If $(i,j) \neq (i'',j'') :$

$$\begin{aligned} S &= \sum_{i'} \left(e^{\frac{2\sqrt{-1}\pi}{ni}(i-i'')} \right)^{i'} \sum_{j'} \left(e^{\frac{2\sqrt{-1}\pi}{nj}(j-j'')} \right)^{j'} \\ &= \frac{1 - \left(e^{\frac{2\sqrt{-1}\pi}{ni}(i-i'')} \right)^{ni}}{1 - e^{\frac{2\sqrt{-1}\pi}{ni}(i-i'')}} \times \frac{1 - \left(e^{\frac{2\sqrt{-1}\pi}{nj}(j-j'')} \right)^{nj}}{1 - e^{\frac{2\sqrt{-1}\pi}{nj}(j-j'')}} \\ &= \frac{1 - e^{2\sqrt{-1}\pi(i-i'')}}{1 - e^{\frac{2\sqrt{-1}\pi}{ni}(i-i'')}} \times \frac{1 - e^{2\sqrt{-1}\pi(j-j'')}}{1 - e^{\frac{2\sqrt{-1}\pi}{ni}(j-j'')}} = 0 \end{aligned}$$

  as the sum of a geometric sequence.

We deduce that

$$\mathscr{F}^{-1}(\mathscr{F}(X))_{ijk} = \frac{1}{ni}\frac{1}{nj} \times ni \times nj \times X_{ijk} = X_{ijk}$$

And finally $\mathscr{F}$ is the reciprocal function of $\mathscr{F}^{-1}$.
For more details about the Convolution sub-layer, see Section 3.3.

3.2.2 Bias sub-layer

The bias layer is a 2D convolution with a kernel size of 1. This means that it only performs matrix multiplication on the channels, but pixel by pixel. In other words, it mixes channels via a kernel, but does not allow interaction between pixels.
Precisely,

$$\mathscr{B}^l_\theta(X)_{ijk} = \sum_{k'} X_{ijk} W_{k'k} + B_k$$

3.3 Some details on the convolution sub-layer

In this section, we will specify some details for the convolution layer.

3.3.1 Border issues

Let $W = \mathscr{F}^{-1}(\hat{W})$, we have :

$$\mathscr{C}_\theta^l(\tilde{X}) = \mathscr{F}^{-1}\left(\mathscr{F}(X) \cdot \hat{W}\right) = \tilde{X} \star W$$

with

$$(\tilde{X} \star W)_{ij} = \sum_{i'j'} \tilde{X}_{i-i'[ni],j-j'[nj]} W_{i'j'}$$

In other words, multiplying in Fourier space is equivalent to performing a $\star$ circular convolution in real space.

Remark. *These modulo operations are only natural for periodic images, which is not our case. The discontinuity that appears when we periodize the image causes oscillations on the edges of the filtered images. To limit this problem, we will apply a padding on the image which is the fact to extend the images by adding pixels all around, before performing the convolution. After the convolution, we restrict the image to partially erase the oscillations.*

3.3.2 FFT

To speed up computations, we will use the FFT (Fast Fourier Transform). The FFT is a fast algorithm to compute the DFT. It is recursive : The transformation of a signal of size N is make from the decomposition of two sub-signals of size $N/2$. The complexity of the FFT is $N\log(N)$ whereas the natural algorithm, which is a matrix multiplication, has a complexity of N^2.

3.3.3 Real DFT

In reality, we'll be using a specific implementation of FFT, called RFFT (Real Fast Fourier Transform). In fact, for $\mathscr{F}^{-1}(A)$ to be real if A is a complex-valued matrix, it is necessary that A respects the Hermitian symmetry:

$$A_{i,nj-(j+1)} = \bar{A}_{i,j}$$

In our case, we want $\mathscr{C}_\theta^l(X)$ to be a real image, so $\mathscr{F}(X) \cdot \hat{W}$ must verify Hermitian-symmetry. To do this, we only need to collect half of the Discrete Fourier Coefficients (DFC) and the other half will be deduced by Hermitian symmetry. More precisely, using the specific RFFT implementation, the DFCs are stored in a matrix of size $(ni, nj//2+1)$. Multiplication can then be performed by the $\hat{W}$ kernel, and when the inverse RFFT is performed, the DFCs will be automatically symmetrized. So the Hermitian symmetry of $\mathscr{F}(X) \cdot \hat{W}$ is verified and $\mathscr{C}_\theta^l(X)$ is indeed a real image.

To simplify, let's assume nk=1. Here is a diagram describing this idea:

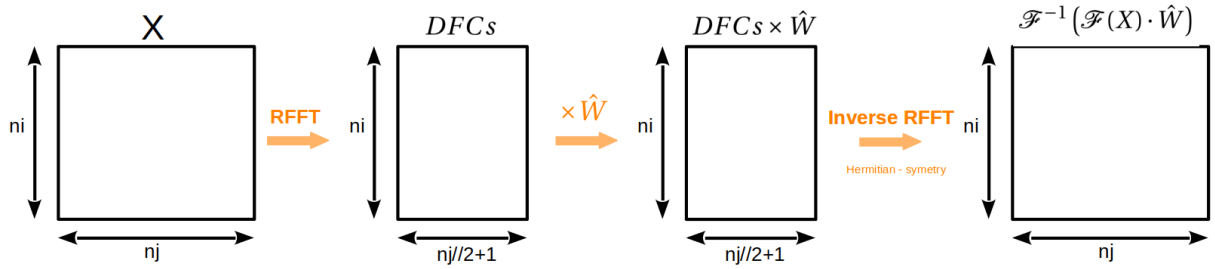

Figure 3.3: RFFT with Hermitian-symmetry scheme.

Remark. *In fact, we can check that $\mathscr{F}(X)$ satisfies Hermitian symmetry immediately.*

3.3.4 Low pass filter

When we perform a DFT on an image, the DFCs related to high frequencies are in practice very low. This is why we can easily filter an image by ignoring these high frequencies, i.e. by truncating the high Fourier modes. In fact, eliminating the higher Fourier modes enables a kind of regularization that helps the generalization. So, in practice, it's sufficient to keep only the DFCs corresponding to low frequencies. Typically, for images of resolution 32×32 to 128×128, we can keep only the 20×20 DFCs associated to low frequencies.
Here is a representation of this idea in 1D :

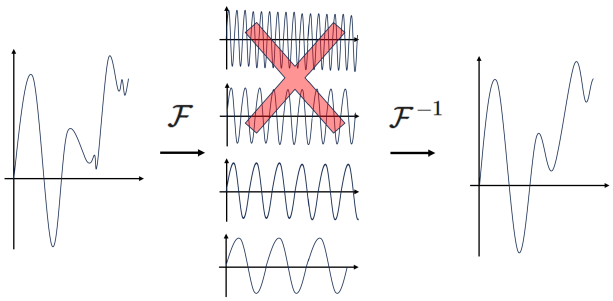

Figure 3.4: Low pass filter.

3.3.5 Global aspect of the FNO

Classical Convolutional Neural Networks (CNN) use very small kernels (typically 3×3). This operation only has a local effect, and it's the sequence of many convolutions that produces more global effects.
In addition, CNNs often use max or mean-pooling layers, which process the image on several scales. Max-pooling (respectively mean-pooling) consists in slicing the image into small pieces of size $n \times n$, then choosing the pixel with the highest value (respectively the average of

the pixels) in each of the small pieces. In most cases, $n = 2$ is used, which divides the number of pixels by 4.
The FNO, on the other hand, uses a $\hat{W}$ frequency kernel and $W = \mathscr{F}^{-1}(\hat{W})$ has full support. For this reason, the effect is immediately non-local. As a result, we can use less layers and we don't need to use a pooling layer.

3.4 Application

In our case, we want to use the FNO to predict the solutions of a PDE. As explained above, we'll train the FNO with a data set (sample of size n_{data}) generated by a ϕ-FEM solver. We will then inject the output of our FNO into a new solver that can correct the solution, i.e. improve its accuracy.
We are still trying to solve the Poisson problem with Dirichlet condition, defined by

$$\begin{cases} -\Delta(\phi w) = f, & \text{on } \Omega, \\ u = g, & \text{in } \Gamma, \end{cases}$$

with $u = \phi w$.
This problem can be approached in different ways. For example, we may want to consider a case where the level-set function ϕ changes, as in the case where we want to solve the problem of the geometry of an organ at different time steps. In this case, we'll need to generate a $\{\phi_i\}_{i=1,\dots,n_{data}}$ collection of level-sets sufficiently representative of the possible variations of the levelset. In a more simple case, if our geometry is defined by an ellipse in a precise domain, the $\{\phi_i\}_{i=1,\dots,n_{data}}$ family will group n_{data} ellipses whose parameters change, such as center or axes.
We may also want to solve the problem for a collection of source terms $\{f_i\}_{i=1,\dots,n_{data}}$. For example, this set could be a Gaussian set whose expected value and variance are varied. In the same idea, we might wish to vary the Dirichlet condition and thus create a collection $\{g_i\}_{i=1,\dots,n_{data}}$.

Remark. *Note that we're working in a discrete way here, so for each i, the terms f_i, g_i and ϕ_i are in fact 2D matrices of size (ni, nj).*

Remark. *Note also that the FNO has less difficulty learning solutions that don't have a wide range of values. This is why the collection of $\{f_i\}$ that we'll be using in the following will in fact be the normalization of the previous collection :*

$$f_{i,norm} = \frac{f_i}{max_{j=1,\dots,n_{data}} ||f_j||_{L^2(\mathscr{O})}}.$$

Here are the steps that will be performed to train the FNO (Figure 3.5):

- We start by creating a dataset containing the level-set, source term and boundary condition collections, defined by

$$X\_train = \{f_i, g_i, \phi_i\}_{i=1,\dots,n_{data}}.$$

Remark. *Note that we can also consider the problem as homogeneous, in which case X_train will only be generated from f and ϕ. We may also wish to fix the geometry, in which case the training sample X_train will not contain the ϕ term.*

We can then solve these n_{data} problems using the ϕ-FEM method, where the solution to each of them is defined by $u = \phi w$. We then define the training sample

$$Y\_train = \{w_i\}_{i=1,\dots,n_{data}}$$

where $u_i = \phi_i w_i$ is the solution associated to the i-th problem of the X_train sample, i.e. solution of

$$\begin{cases} -\Delta(\phi_i w_i) = f_i, & \text{on } \Omega, \\ u_i = g_i, & \text{in } \Gamma. \end{cases}$$

Remark. *Note that in practice, we have enriched the data by increasing the number of channels in the X_train sample. In fact, for each problem i, we add to the sample the primitives of f_i according to x and y, as well as the second primitives according to xx and yy. The X_train sample is then of size (n_{data}, ni, nj, nk) with $nk = 7$ the number of channels if we consider the 3 collections. The Y_train sample is of size $(n_{data}, ni, nj, 1)$.*

- At this moment, we have the (X_train,Y_train) pair that will enable us to train our FNO. More precisely, we're looking to minimize a loss function on the model's θ parameters by using gradient descent. We'll choose the following loss function:

$$loss_\theta = loss_\theta^{(0)} + loss_\theta^{(1)} + loss_\theta^{(2)}$$

with

$$loss_\theta^{(0)} = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} mse(\phi_i w_i - \phi_i w_{\theta,i})$$

$$loss_\theta^{(1)} = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} mse(\nabla_x(\phi_i w_i) - \nabla_x(\phi_i w_{\theta,i})) + mse(\nabla_y(\phi_i w_i) - \nabla_y(\phi_i w_{\theta,i}))$$

$$loss_\theta^{(2)} = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} mse(\nabla_{xx}(\phi_i w_i) - \nabla_{xx}(\phi_i w_{\theta,i})) + mse(\nabla_{yy}(\phi_i w_i) - \nabla_{yy}(\phi_i w_{\theta,i}))$$

with w_i the ϕ-FEM solution associated to the i-th problem considered (i.e. the i-th Y_train sample data), $w_{\theta,i}$ the FNO prediction associated to the i-th problem, ϕ_i the i-th level-set and mse the "Mean Square Error" function defined by

$$mse(A) = \frac{1}{ni}\frac{1}{nj} \sum_{i=0}^{ni-1} \sum_{j=0}^{nj-1} A_{i,j}^2.$$

Remark. *Note that first and second derivatives according to x and y are calculated here by finite differences.*

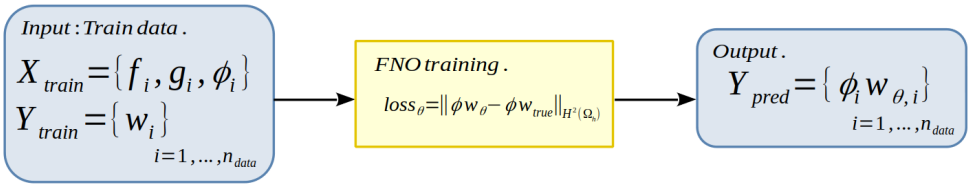

Figure 3.5: Representation of the FNO training.

We can now proceed to the correction stage (Figure 3.6), where we consider a type of correction defined in Section 4.2 (correction by adding, correction by multiplying or correction by multiplying on an enhanced problem). We can then consider a new test sample X_test constructed in the same way as the training sample and defined by

$$X\_test = \{f_i^{test}, g_i^{test}, \phi_i^{test}\}_{i=1,\dots,n_{test}}.$$

We will then provide this sample as input to the FNO in order to obtain its prediction $w_{\theta,i}^{test}$ for each problem i of the test sample (where θ corresponds to the parameters learned during training). We then construct $u_{\theta_i}^{test} = \phi_i w_{\theta,i}^{test}$ which will be given as input to one of the correction solvers. We will then obtain what we call the corrected solution.

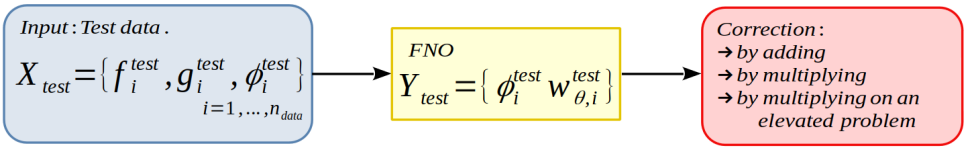

Figure 3.6: Representation of correction steps

4 Correction

In this section, we'll introduce the notion of correction, with the aim of correcting the output of neural networks (in particular the FNO), which are extremely fast but not very accurate. Thus, we will look at the different correction methods. We will start by presenting the different problems considered (Section 4.1), then we will introduce the different correction methods (Section 4.2), more specifically correction by addition, correction by multiplication and correction by multiplication on a so-called elevated problem. We will then continue by presenting some theoretical results on these different methods (Section 4.3), and finish by presenting the numerical results obtained (Section 4.4).

4.1 Presentation of different problems considered

In this section, we present the different problems we will be considering in Section 4.4. We will consider the geometry of a circle in Section 4.1.1 represented in Figure 4.1 as well as the geometry of a square in Section 4.1.2 represented in Figure 4.2. For the circle, we will consider a first analytic trigonometric solution, parameterized in such a way that the problem can be considered homogeneous. we will then consider a second problem for which no exact solution is known, and for which we will take an over-refined FEM solution as the reference solution. For the square, we will consider only an analytic trigonometric solution parameterized in such a way that the problem can be considered homogeneous.

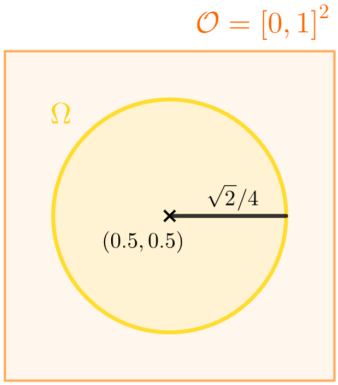

Figure 4.1: Representation of the first domain considered : the Circle.

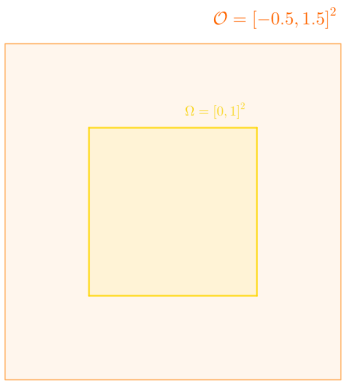

Figure 4.2: Representation of the second domain considered : the Square.

Remark. *To generate meshes with FEM, we will consider a mesh comparable to ϕ-FEM. To do this, we generate our ϕ-FEM grid, which is simply the regular mesh of the $\mathcal{O}$ domain of $n_{vert} \times n_{vert}$ nodes. From this mesh we generate a FEM mesh whose largest element diameter h_{FEM} is very close to the largest diameter associated with $\phi - FEM$, $h_{\phi-FEM}$ ($h_{FEM} \sim h_{\phi-FEM}$). Here, we have a representation of the meshes generated for FEM and ϕ-FEM on the circle (Figure 4.3) and on the square (Figure 4.4) with $n_{vert} = 10$ vertices in each direction.*

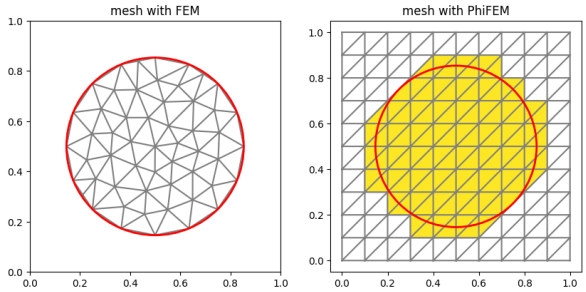

Figure 4.3: Representation of meshes for FEM and ϕ-FEM on the Circle.

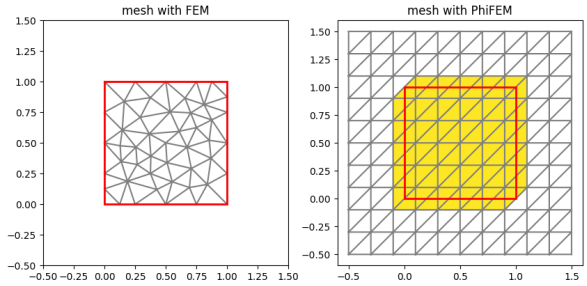

Figure 4.4: Representation of meshes for FEM and ϕ-FEM on the Square.

Remark. *Note that in the case where the problem is non-homogeneous, if we don't have a definition of the Dirichlet boundary condition, we can consider*

$$g(x,y) = u_{ex}(x,y) \times (1+\phi(x,y))$$

with ϕ the level-set function, null by definition on Γ.

4.1.1 First domain : the Circle.

Here, we will consider the Ω domain to be the circle of radius $\sqrt{2}/4$ and center $(0.5, 0.5)$.This domain is entirely included in the fictitious domain $O = [0,1]^2$ (Figure 4.1).
We will consider the level-set function ϕ defined by

$$\phi(x,y) = -1/8 + (x-1/2)^2 + (y-1/2)^2,$$

negative function inside the domain, null at the boundary and positive outside it
We will consider a first analytic trigonometric solution (Section 4.1.1.1), parameterized in such a way that the problem can be considered homogeneous and then consider a second problem (Section 4.1.1.2) for which no exact solution is known, and for which we'll take an over-refined FEM solution as the reference solution.

4.1.1.1 First problem

Here, we are interested in the Poisson problem with an analytical solution

$$u_{ex}(x,y) = S \times \sin\left(8\pi f\left((x-0.5)^2 + (y-0.5)^2\right) + \varphi\right)$$

where $S \in [0,1]$ is the amplitude of the signal, $f \in \mathbb{N}$ can be associated with the "frequency" of the signal and $\varphi \in [0,1]$ the phase at the origin, represented in Figure 4.5.
Thus, the second associated member is defined by

$$\begin{aligned} f(x,y) = 256\pi^2 S f^2 \left((x-0.5)^2 + (y-0.5)^2\right) \sin\left[8\pi f\left((x-0.5)^2 + (y-0.5)^2\right) + \varphi\right] \\ - 32\pi S f \cos\left[8\pi f\left((x-0.5)^2 + (y-0.5)^2\right) + \varphi\right] \end{aligned}$$

and represented in Figure 4.6.

Remark. *Note that for $\varphi = 0$, the Dirichlet conditions considered are then homogeneous on the circle (i.e. $g = 0$ on Γ).*

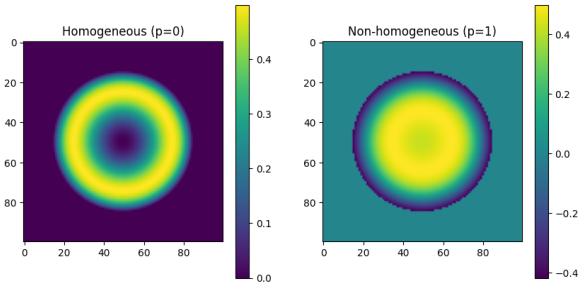

Figure 4.5: Representation of the solution u_{ex} on the Circle with $S = 0.5$ and $f = 1$ (with $p = 0$ and $p = 1$).

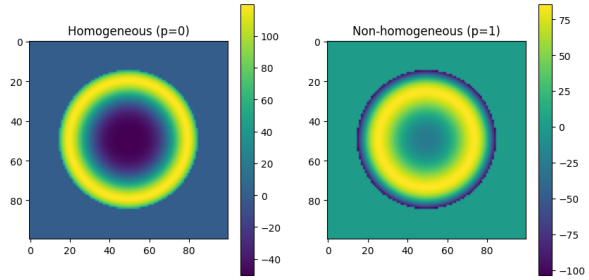

Figure 4.6: Representation of the term f on the Circle with $S = 0.5$ and $f = 1$ (with $p = 0$ and $p = 1$).

4.1.1.2 f Gaussian

In the case considered here, no analytical solution is known. We consider the source term f to be a Gaussian defined by

$$f(x,y) = \exp\left(-\frac{(x-\mu_0)^2 + (y-\mu_1)^2}{2\sigma^2}\right),$$

with $\sigma \sim \mathcal{U}([0.1, 0.6])$ and $\mu_0, \mu_1 \sim \mathcal{U}([0.5 - \sqrt{2}/4, 0.5 + \sqrt{2}/4])$ with the condition $\phi(\mu_0, \mu_1) < -0.05$ and represented in Figure 4.7

Since we don't have an exact solution, we will consider a so-called reference solution, denoted u_{ref}. This reference solution is defined as an over-refined $\mathbb{P}^1$ solution obtained by the standard FEM method (with $h_{ref} \approx 0.006$). It is to this reference solution that we can compare the solutions obtained by the correction methods or by the FNO.

Remark. *For this problem, we choose to consider only the homogeneous case, i.e. we take $g = 0$ on Γ.*

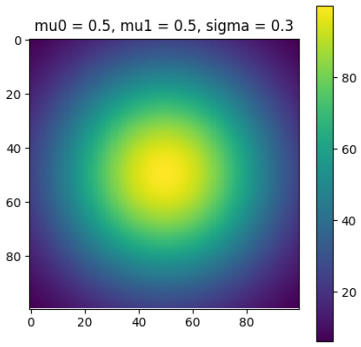

Figure 4.7: Representation of the term f on the Circle with $(\mu_0, \mu_1) = (0.5, 0.5)$ and $\sigma = 0.3$.

4.1.2 Second domain : the Square.

Here, we will consider the Ω domain to be the unit square $\Omega = [0,1]^2$. This domain is entirely included in the fictitious domain $O = [-0.5, 1.5]^2$ (Figure 4.2).
In this case, we will consider two level-set functions. The first function ϕ, defined by

$$\phi(x,y) = x(1-x)y(1-y),$$

will be used when solving the weak problem. However, it cannot be used to construct our cell and face sets, because it is not only negative in the Ω domain (Figure 4.8).

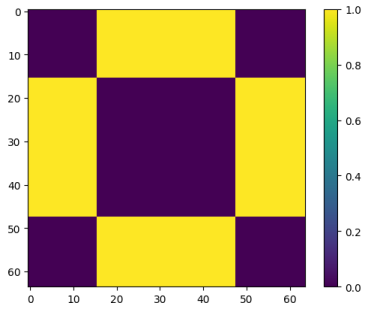

Figure 4.8: Representation of $\phi(x,y) < 0$.

To construct the sets of cells and faces needed to perform the ϕ-FEM method, we will consider a second function, denoted ϕ_C and defined by

$$\phi_C(x,y) = \max(|x-0.5|, |y-0.5|) - 0.5,$$

which is indeed negative inside the domain, zero at the boundary and positive outside, but which does not sufficiently satisfy the regularity conditions necessary for ϕ-FEM.
We will consider only an analytic trigonometric solution parameterized in such a way that the problem can be considered homogeneous.

4.1.2.1 Problem

Here, we are interested in the Poisson problem with as analytical solution

$$u_{ex}(x,y) = S \times \sin\left(2\pi f x + \varphi\right) \times \sin\left(2\pi f y + \varphi\right)$$

where $S \in [0,1]$ is the amplitude of the signal, $f \in \mathbb{N}$ can be associated with the "frequency" of the signal and $\varphi \in [0,1]$ the phase at the origin.
Thus, the associated second member is defined by

$$f(x,y) = 8\pi^2 S f^2 \sin\left(2\pi f x + \varphi\right) \sin\left(2\pi f y + \varphi\right)$$

Remark. *It should be noted that as for the circle for $\varphi = 0$, the Dirichlet conditions considered are then homogeneous on the square (i.e. $g = 0$ on Γ).*

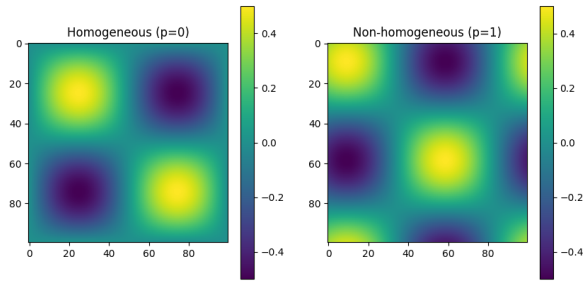

Figure 4.9: Representation of the solution u_{ex} on the Circle with $S = 0.5$ and $f = 1$ (with $p = 0$ and $p = 1$).

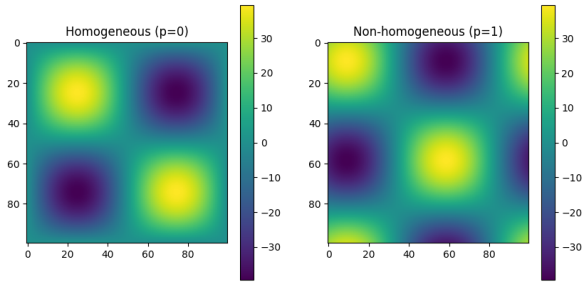

Figure 4.10: Representation of the term f on the Circle with $S = 0.5$ and $f = 1$ (with $p = 0$ and $p = 1$).

4.2 Presentation of the different correction methods considered

Here we are given $\tilde{\phi}$ an "initial" solution to the problem under consideration, i.e. a solution that has not yet been corrected. This may be a perturbed analytic solution, a ϕ-FEM solution, or a solution predicted by a neural network (such as an FNO, a Multi-perceptron network or PINNs, for example). The aim is to inject this solution into a new problem in order to improve the accuracy of the solution. To achieve this, we consider 3 types of correction: correction by addition (Section 4.2.1), correction by multiplication (Section 4.2.2) and correction by multiplication on an elevated problem (Section 4.2.3).

Remark. *In what follows, we assume that $\tilde{\phi}$ already has the right conditions at the boundary, i.e. $\tilde{\phi} = g$ on Γ.*

4.2.1 Correction by adding

In this first method, we will try to approximate the solution obtained $\tilde{\phi}$ to the exact solution by completing the difference between the two, which is what we will call correction by adding. To do this, we will consider

$$\tilde{u} = \tilde{\phi} + \tilde{C}$$

and we want to find $\tilde{C} : \Omega \to \mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta \tilde{u} = f, & \text{on } \Omega, \\ \tilde{u} = g, & \text{in } \Gamma. \end{cases}$$

Remark. *Note that this problem is in fact equivalent to the initial problem. We only hope that the approximate solution $\tilde{u}$ obtained is more accurate than the approximate solution u obtained by solving the initial problem.*

Rewriting the problem, we seek to find $\tilde{C} : \Omega \to \mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta \tilde{C} = \tilde{f}, & \text{on } \Omega, \\ \tilde{C} = 0, & \text{in } \Gamma. \end{cases}$$

with $\tilde{f} = f + \Delta\tilde{\phi}$.
Thus for the standard FEM method, the weak formulation will be given by

$$\int_\Omega \nabla\tilde{C}\cdot\nabla v = \int_\Omega \tilde{f} v$$

where the homogeneous Dirichlet conditions can be strongly imposed by classical methods (penalization, elimination...).
For the ϕ-FEM method, we look for C such that $\tilde{C} = \phi C$ and the weak formulation (associated with a homogeneous problem because $\tilde{C} = 0$ on Γ) is given by

$$\int_{\Omega_h} \nabla(\phi C)\cdot\nabla(\phi v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\phi C)\phi v + G_h(w, v) = \int_{\Omega_h} \tilde{f}\phi v + G_h^{rhs}(v)$$

with

$$G_h(C, v) = \sigma h \sum_{E\in\mathscr{F}_h^\Gamma} \int_E \left[\frac{\partial}{\partial n}(\phi C)\right]\left[\frac{\partial}{\partial n}(\phi v)\right] + \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma} \int_T \Delta(\phi C)\Delta(\phi v)$$

and

$$G_h^{rhs}(v) = -\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma} \int_T \tilde{f}\Delta(\phi v).$$

In the non-homogeneous case, the formulation is the same, as the correction problem stays homogeneous ($\tilde{C} = 0$ on Γ).

Remark. *In practice, it may be useful to integrate by parts the term containing $\Delta\tilde{\phi}$ (implicitly included in $\tilde{f}$).*
So for FEM, as $v \in H_0^1(\Omega)$, we have

$$\int_\Omega \tilde{f} v = \int_\Omega f v + \int_\Omega \Delta\tilde{\phi} v = \int_\Omega f v - \int_\Omega \nabla\tilde{\phi}\cdot\nabla v.$$

For ϕ-FEM, we have

$$\int_{\Omega_h} \tilde{f}\phi v = \int_{\Omega_h} f\phi v + \int_{\Omega_h} \Delta\tilde{\phi}\phi v = \int_{\Omega_h} f\phi v - \int_{\Omega_h} \nabla\tilde{\phi}\cdot\nabla(\phi v) + \int_{\partial\Omega_h} \frac{\partial\tilde{\phi}}{\partial n}\phi v.$$

4.2.2 Correction by multiplying

In this second method, we try to approach the exact solution in a different way. In fact, we want to bring the factor between the $\tilde{\phi}$ solution and the solution of the corrected problem closer to 1. In other words, by considering

$$\tilde{u} = \tilde{\phi} C,$$

we try to bring $C = \frac{\tilde{u}}{\tilde{\phi}}$ closer to 1 (for $\tilde{\phi} \neq 0$). This type of correction is called correction by multiplying.
So we're looking for $C : \Omega \to \mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta(\tilde{\phi} C) = f, & \text{on } \Omega, \\ C = 1, & \text{on } \Gamma. \end{cases}$$

Remark. *In the same way as for correction by adding, we note that this problem is equivalent to the initial problem.*

So for the standard FEM method, the weak formulation will be given by

$$\int_\Omega \nabla(\tilde{\phi}C)\cdot\nabla(\tilde{\phi}v) = \int_\Omega f\tilde{\phi}v$$

where homogeneous or non-homogeneous Dirichlet conditions can be strongly imposed by classical methods (penalization, elimination...).
For the ϕ-FEM method, the weak formulation for the homogeneous problem is given by

$$\int_{\Omega_h} \nabla(\tilde{\phi}C)\cdot\nabla(\tilde{\phi}v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\tilde{\phi}C)\tilde{\phi}v + G_h(w,v) = \int_{\Omega_h} f\tilde{\phi}v + G_h^{rhs}(v)$$

with

$$G_h(C,v) = \sigma h \sum_{E\in\mathscr{F}_h^\Gamma}\int_E \left[\frac{\partial}{\partial n}(\tilde{\phi}C)\right]\left[\frac{\partial}{\partial n}(\tilde{\phi}v)\right] + \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T \Delta(\tilde{\phi}C)\Delta(\tilde{\phi}v)$$

and

$$G_h^{rhs}(v) = -\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T f\Delta(\tilde{\phi}v).$$

In the non-homogeneous case, it is important to impose the boundary conditions either by the direct method or by the dual method presented in Section 2.2.3.2.

4.2.3 Correction by multiplying on an elevated problem

We now introduce a third correction method, which we will call correction by multiplying on an elevated problem. This method is in fact very similar to the previous one (correction by multiplying), except that we are no longer trying to correct the same problem.
The initial modified problem, which we now consider, consists in finding $u:\Omega\to\mathbb{R}^d$ such that

$$\begin{cases} -\Delta\hat{u} = f, & \text{in } \Omega, \\ \hat{u} = g + m, & \text{on } \Gamma, \end{cases}$$

with $\hat{u} = u + m$ and m a constant.
We then apply the same multiplication correction method, but this time on the modified problem, which has been elevated by a constant m. We then consider

$$\tilde{u} = \hat{\phi}C$$

with

$$\hat{\phi} = \tilde{\phi} + m$$

and so we look for $C:\Omega\to\mathbb{R}^d$ solution to the problem

$$\begin{cases} -\Delta(\hat{\phi}C) = f, & \text{in } \Omega, \\ C = 1, & \text{on } \Gamma. \end{cases}$$

So for the standard FEM method, the weak formulation will be given by

$$\int_\Omega \nabla(\hat{\phi}C)\cdot\nabla(\hat{\phi}v) = \int_\Omega f\hat{\phi}v$$

where homogeneous or non-homogeneous Dirichlet conditions can be strongly imposed by classical methods (penalization, elimination...)
For the ϕ-FEM method, the weak formulation for the homogeneous problem is given by

$$\int_{\Omega_h} \nabla(\hat{\phi}C)\cdot\nabla(\hat{\phi}v) - \int_{\partial\Omega_h} \frac{\partial}{\partial n}(\hat{\phi}C)\hat{\phi}v + G_h(w,v) = \int_{\Omega_h} f\hat{\phi}v + G_h^{rhs}(v)$$

with

$$G_h(C,v) = \sigma h \sum_{E\in\mathscr{F}_h^\Gamma}\int_E \left[\frac{\partial}{\partial n}(\hat{\phi}C)\right]\left[\frac{\partial}{\partial n}(\hat{\phi}v)\right] + \sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T \Delta(\hat{\phi}C)\Delta(\hat{\phi}v)$$

and

$$G_h^{rhs}(v) = -\sigma h^2 \sum_{T\in\mathscr{T}_h^\Gamma}\int_T f\Delta(\hat{\phi}v).$$

In the case of this correction, whether the problem is homogeneous or non-homogeneous, the correction problem is non-homogeneous (for $m \neq 0$), so it is important to impose the boundary conditions either by the direct method, or by the dual method presented in Section 2.2.3.2.

Remark. *Note that if the problem is sufficiently elevated for its solution to be strictly positive, the operation of bringing $C = \frac{\tilde{u}}{\hat{\phi}}$ closer to 1 doesn't pose any problem (since in this case $\hat{\phi} \neq 0$). Moreover, we can easily return to the original problem by subtracting m from $\tilde{u}$.*

4.3 Theoretical results

The aim of this section is to present some interesting theoretical results. We begin by presenting the interest of elevating the problem in Section 4.3.1. We will then present an interesting result for standard FEM in Section 4.3.2. In fact, we'll show that the solution obtained by the multiplication correction on the elevated problem converges to the solution obtained by the addition correction when m tends to infinity. Finally, we will demonstrate the estimation error of the correction on the elevated problem in Section 4.3.3.

4.3.1 Interest of elevating the problem

The aim of this section is to explain the interest in elevating the problem in the case of the multiplication correction presented in Section 4.2.3 with a $\mathbb{P}^1$ solution. Only the standard FEM method is considered here. For this purpose, we will consider a so-called disturbed solution, i.e. one that is close to the exact solution. We assume we have an analytical solution to the problem, denoted u_{ex}, to which we will apply a perturbation, denoted P. We then define

$$\tilde{\phi}(x,y) = u(x,y) + \epsilon P(x,y)$$

with P the perturbation (such that $P = 0$ on Γ) and ϵ small.
We pose

$$\hat{\phi} = \tilde{\phi} + m$$

with m a constant, and consider the multiplication correction on a elevated problem, defined in Section 4.2.3 by the problem

$$\begin{cases} -\Delta(\hat{\phi}C) = f, & \text{in } \Omega, \\ \hat{u} = g + m, & \text{on } \Gamma. \end{cases}$$

with $\hat{u} = \hat{\phi}C$.
In Section 4.3.3, we will prove the following inequality in the context of this problem

$$||\hat{u_{ex}} - \hat{u_h}||_0 \leq ch^{k+1}||\hat{\phi}||_\infty |C|_{k+1,\Omega}$$

with c a constant, $\hat{u_{ex}} = u_{ex} + m$ and $C = \frac{u_{ex}+m}{\hat{\phi}}$.
For $k = 1$, we then have

$$|C|_{2,\Omega} = \left|\frac{u_{ex}+m}{\hat{\phi}}\right|_{2,\Omega} = \left|\left|\left(\frac{u_{ex}+m}{\hat{\phi}}\right)''\right|\right|_{0,\Omega} = \epsilon\left|\left|\left(\frac{P}{\hat{\phi}}\right)''\right|\right|_{0,\Omega} = \epsilon\left|\left|\left(\frac{P}{\tilde{\phi}+m}\right)''\right|\right|_{0,\Omega}$$

with

$$\begin{aligned}\left(\frac{P}{\tilde{\phi}+m}\right)'' &= \frac{P''(\tilde{\phi}+m) - P\tilde{\phi}''}{(\tilde{\phi}+m)^2} + \frac{2(P\tilde{\phi}' - P'(\tilde{\phi}+m))\tilde{\phi}'}{(\tilde{\phi}+m)^3} \\ &= \frac{P''\tilde{\phi} - P\tilde{\phi}''}{(\tilde{\phi}+m)^2} + \frac{2(P\tilde{\phi}' - P'\tilde{\phi})\tilde{\phi}'}{(\tilde{\phi}+m)^3} + \frac{mP''}{(\tilde{\phi}+m)^2} - \frac{2mP'\tilde{\phi}'}{(\tilde{\phi}+m)^3}\end{aligned}$$

As

$$\left|\left|\left(\frac{P}{\tilde{\phi}+m}\right)''\right|\right|_{0,\Omega} = \left|\left|\left(\frac{P}{m\left(1+\frac{\tilde{\phi}}{m}\right)}\right)''\right|\right|_{0,\Omega} = \frac{1}{m}\left|\left|\left(\frac{P}{1+\frac{\tilde{\phi}}{m}}\right)''\right|\right|_{0,\Omega}$$

and

$$\begin{aligned}\left(\frac{P}{1+\frac{\tilde{\phi}}{m}}\right)'' = m\left(\frac{P}{\hat{\phi}}\right)'' &= \frac{m(P''\tilde{\phi} - P\tilde{\phi}'')}{(\tilde{\phi}+m)^2} + \frac{2m(P\tilde{\phi}' - P'\tilde{\phi})\tilde{\phi}'}{(\tilde{\phi}+m)^3} + \frac{m^2P''}{(\tilde{\phi}+m)^2} - \frac{2m^2P'\tilde{\phi}'}{(\tilde{\phi}+m)^3} \\ &= \frac{m(P''\tilde{\phi} - P\tilde{\phi}'')}{m^2\left(1+\frac{\tilde{\phi}}{m}\right)^2} + \frac{2m(P\tilde{\phi}' - P'\tilde{\phi})\tilde{\phi}'}{m^3\left(1+\frac{\tilde{\phi}}{m}\right)^3} + \frac{m^2P''}{m^2\left(1+\frac{\tilde{\phi}}{m}\right)^2} - \frac{2m^2P'\tilde{\phi}'}{m^3\left(1+\frac{\tilde{\phi}}{m}\right)^3} \\ &= \frac{P''\tilde{\phi} - P\tilde{\phi}''}{m\left(1+\frac{\tilde{\phi}}{m}\right)^2} + \frac{2(P\tilde{\phi}' - P'\tilde{\phi})\tilde{\phi}'}{m^2\left(1+\frac{\tilde{\phi}}{m}\right)^3} + \frac{P''}{\left(1+\frac{\tilde{\phi}}{m}\right)^2} - \frac{2P'\tilde{\phi}'}{m\left(1+\frac{\tilde{\phi}}{m}\right)^3}\end{aligned}$$

then, for m sufficiently large :

$$\left|\left|\left(\frac{P}{1+\frac{\tilde{\phi}}{m}}\right)''\right|\right|_{0,\Omega} \sim \left|\left|P''\right|\right|_{0,\Omega}$$

Finally, for m sufficiently large, we have $||\hat{\phi}||_\infty \sim m$ and thus

$$\left|\left|\frac{u_{ex}+m}{\hat{\phi}} - C_h\right|\right|_{0,\Omega} \leq ch^{k+1}\epsilon\left|\left|P''\right|\right|_{0,\Omega}$$

We deduce that when m is big, the error no longer depends on the solution but only on the perturbation P. These results were obtained numerically and are presented in Section 4.4.2. However, when m is small, we can see that the error is dominated by the first derivatives and second derivatives of the disturbed solution $\tilde{\phi}$.

Remark. *In addition, raising the problem can be extremely efficient when the solution cancels out on the domain. However, note that in the case of the addition correction presented in Section 4.2.1, elevated the problem is of no interest.*

4.3.2 Comparison of correction methods

Here, we compare 2 methods, the addition correction method defined in Section 4.2.1 and the multiplication correction method on a elevated problem defined in Section 4.2.3. In fact, we will show that when m tends to infinity, the solution of the multiplication correction on a elevated problem converges to the solution of the addition correction.

Correction by adding :

Let's start by looking at the form of the solution in the case of correction by addition. The decomposition of u_h on the $(\varphi_1,\ldots,\varphi_{N_h})$ basis of V_h is written as follows for this problem

$$u_h = C_h + \tilde{\phi}(x) = \left(\sum_{i=1}^{N_h} C_i\varphi_i\right) + \tilde{\phi}(x) \tag{3}$$

We have

$$C_i = u_{ex}(x_i) - \tilde{\phi}(x_i) \tag{4}$$

with

$$u_{ex}(x_i) = \tilde{\phi}(x_i) - \epsilon P(x_i) \tag{5}$$

With the 2 previous relations, we can develop 3 :

$$\begin{aligned} u_h &= \tilde{\phi}(x) + \sum_{i=1}^{N_h} C_i\varphi_i \\ &= \tilde{\phi}(x) + \sum_{i=1}^{N_h} (u(x_i) - \tilde{\phi}(x_i))\varphi_i \quad \text{by 4} \end{aligned}$$

$$= \tilde{\phi}(x) + \sum_{i=1}^{N_h} (\tilde{\phi}(x_i) - \epsilon P(x_i) - \tilde{\phi}(x_i))\varphi_i \quad \text{by 5}$$

$$u_h = \tilde{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i)\varphi_i \tag{6}$$

Correction by multiplying on an elevated problem :
We are now interested in the form of the solution in the case of multiplication correction on a elevated problem. The decomposition of $\hat{u_h}$ on the $(\varphi_1, \ldots, \varphi_{N_h})$ basis of V_h is written as follows for this problem

$$\hat{u_h} = C_h \hat{\phi} = \left(\sum_{i=1}^{N_h} C_i \varphi_i \right) \hat{\phi}(x) \tag{7}$$

We have

$$C_i = \frac{u(x_i) + m}{\hat{\phi}(x_i)} = \frac{u(x_i) + m}{\tilde{\phi}(x_i) + m} \tag{8}$$

with

$$u(x_i) = \tilde{\phi}(x_i) - \epsilon P(x_i) \tag{9}$$

and

$$\tilde{\phi}(x) = \tilde{\phi}(x_i) + (x - x_i)\tilde{\phi}'(x_i). \tag{10}$$

Moreover, we have

$$\sum_{i=1}^{N_h} \varphi_i = 1 \tag{11}$$

With the 4 previous relations, we can develop 7 :

$$\begin{aligned}
\hat{u_h} &= \left(\sum_{i=1}^{N_h} C_i \varphi_i \right) \hat{\phi}(x) \\
&= \left(\sum_{i=1}^{N_h} \frac{u(x_i) + m}{\tilde{\phi}(x_i) + m} \varphi_i \right) \hat{\phi}(x) \quad \text{by 8} \\
&= \left(\sum_{i=1}^{N_h} \frac{\tilde{\phi}(x_i) + m - \epsilon P(x_i)}{\tilde{\phi}(x_i) + m} \varphi_i \right) \hat{\phi}(x) \quad \text{by 9} \\
&= \sum_{i=1}^{N_h} \left(1 - \epsilon \frac{P(x_i)}{\tilde{\phi}(x_i) + m} \right) \varphi_i \hat{\phi}(x) \\
&= \left(\sum_{i=1}^{N_h} \varphi_i \right) \hat{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i) \frac{\hat{\phi}(x)}{\tilde{\phi}(x_i) + m} \varphi_i \\
&= \hat{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i) \frac{\tilde{\phi}(x_i) + m + (x - x_i)\tilde{\phi}'(x_i)}{\tilde{\phi}(x_i) + m} \varphi_i \quad \text{by 10 et 11} \\
&= \hat{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i) \left(1 + \frac{(x - x_i)\tilde{\phi}'(x_i)}{\tilde{\phi}(x_i) + m} \right) \varphi_i
\end{aligned}$$

$$= \tilde{\phi}(x) + m - \epsilon \sum_{i=1}^{N_h} P(x_i) \left(1 + \frac{(x - x_i)\tilde{\phi}'(x_i)}{\tilde{\phi}(x_i) + m}\right) \varphi_i$$

Thus

$$u_h = \hat{u_h} - m = \tilde{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i) \left(1 + \frac{(x - x_i)\tilde{\phi}'(x_i)}{\tilde{\phi}(x_i) + m}\right) \varphi_i$$

and finally

$$u_h \xrightarrow[m \to \infty]{} \tilde{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i)\varphi_i \tag{12}$$

So by 6 and 12, it would seem that the 2 proposed methods are equivalent (taking m to be large).

Remark. *Taking $m = 0$, we return to the case of correction by multiplication defined in Section 4.2.2 and the solution is of the form*

$$u_h = \tilde{\phi}(x) - \epsilon \sum_{i=1}^{N_h} P(x_i) \left(1 + \frac{(x - x_i)\tilde{\phi}'(x_i)}{\tilde{\phi}(x_i)}\right) \varphi_i$$

4.3.3 Error estimation of the correction on the elevated problem

As in the Section 4.3.1, we assume we have an analytical solution to the problem, denoted u_{ex}, to which we will apply a perturbation, denoted P. We then define

$$\tilde{\phi}(x, y) = u(x, y) + \epsilon P(x, y)$$

with P the perturbation (such that $P = 0$ on Γ) and ϵ small.
We consider

$$\hat{\phi} = \tilde{\phi} + m = u_{ex} + \epsilon P + m = \hat{u_{ex}} + \epsilon P$$

with $\hat{u_{ex}} = u_{ex} + m$ et m a constant.
We still want to solve the following problem:

$$\begin{cases} -\Delta(\hat{\phi}C) = f, & \text{in } \Omega, \\ \hat{u} = g + m, & \text{on } \Gamma. \end{cases}$$

with $\hat{u} = \hat{\phi}C$ whose approximate variational problem consists of

$$\text{Find } \hat{u_h} \in V_h \text{ such that } a(\hat{u_h}, v_h) = l(v_h),\ \forall v_h \in V_h$$

Here we seek to prove the following property:

$$||\hat{u} - \hat{u_h}||_0 \leq c h^{k+1} ||\hat{\phi}||_\infty |C|_{k+1}$$

- We're first interested in $||\hat{u} - \hat{u_h}||_1$.

  Since V_h is a vector subspace of V, posing $v = v_h$, we obtain :

  $$a(\hat{\phi}C, \hat{\phi}v_h) - a(\hat{\phi}C_h, \hat{\phi}v_h) = 0 \quad \forall v_h \in V_h.$$

  We then have Galerkin orthogonality:

  $$a(\hat{u} - \hat{u_h}, v_h) = 0 \quad \forall v_h \in V_h$$

  and

  $$\begin{aligned}
  ||\hat{u} - \hat{u_h}||_1^2 &\leq \alpha\, a(\hat{u} - \hat{u_h}, \hat{u} - \hat{u_h}) && \text{by coercivity} \\
  &= \alpha\, a(\hat{u} - \hat{u_h}, \hat{u} - I_h\hat{u} + I_h\hat{u} - \hat{u_h}) \\
  &= \alpha\, a(\hat{u} - \hat{u_h}, \hat{u} - I_h\hat{u}) && \text{by Galerking orthogonality taking } v_h = \hat{u_h} - I_h\hat{u} \\
  &\leq \alpha |\hat{u} - \hat{u_h}|_1 |\hat{u} - I_h\hat{u}|_1 && \text{by continuity} \\
  &\leq \alpha ||\hat{u} - \hat{u_h}||_1 |\hat{u} - I_h\hat{u}|_1.
  \end{aligned}$$

  Thus

  $$||\hat{u} - \hat{u_h}||_1 \leq \alpha |\hat{u} - I_h\hat{u}|_1 = \alpha |(C - I_hC)\hat{\phi}|_1.$$

  By posing $A = C - I_hC$, we have

  $$|A\hat{\phi}|_1 = ||(A\hat{\phi})'||_0 = ||A'\hat{\phi} + A\hat{\phi}'||_0 \leq ||A'\hat{\phi}||_0 + ||A\hat{\phi}'||_0 \leq \alpha ||\hat{\phi}||_\infty ||A||_1$$

  because

  $$||A'\hat{\phi}||_0 = \sqrt{\int_\Omega (A'\hat{\phi})^2} \leq \max_\Omega \hat{\phi} \sqrt{\int_\Omega (A')^2} = ||\hat{\phi}||_\infty |A|_1 \leq ||\hat{\phi}||_\infty ||A||_1$$

  and

  $$||A\hat{\phi}'||_0 = \sqrt{\int_\Omega (A\hat{\phi}')^2} \leq \max_\Omega \hat{\phi}' \sqrt{\int_\Omega (A)^2} = ||\hat{\phi}'||_\infty ||A||_0 \leq \alpha ||\hat{\phi}||_\infty ||A||_1.$$

  Therefore, we have

  $$|\hat{u} - I_h\hat{u}|_1 = |(C - I_hC)\hat{\phi}|_1 \leq \alpha ||\hat{\phi}||_\infty ||C - I_hC||_1$$

  Finally, using the interpolation inequality, we obtain

  $$||\hat{u} - \hat{u_h}||_1 \leq \alpha h^k ||\hat{\phi}||_\infty |C|_{k+1} \tag{13}$$

- We can now turn our attention directly to $||\hat{u} - \hat{u_h}||_0$.

  We'll start by applying the Aubin-Nitsche duality method by considering the dual problem:

Let $\hat{z} \in H_0^1(\Omega)$ solution of the problem

$$\begin{cases} -\Delta \hat{z} = \hat{e_h}, & \text{in } \Omega \\ \hat{z} = 0, & \text{on } \Gamma \end{cases}$$

with $\hat{e_h} = \hat{u} - \hat{u_h}$. Thus

$$a(u,v) = -\int_\Omega \Delta u \cdot v = \int_\Omega \nabla u \cdot \nabla v$$

and as $\hat{e_h} \in H_0^1(\Omega)$

$$a(\hat{z}, \hat{e_h}) = \int_\Omega (-\Delta \hat{z}) \cdot \hat{e_h} = \int_\Omega \hat{e_h}^2 = ||\hat{e_h}||_0^2 \tag{14}$$

Moreover, by the regularity properties :

$$\hat{z} \in H^2(\Omega) \tag{15}$$

and

$$||\hat{z}||_2 \leq \alpha ||\hat{e_h}||_0 = \alpha ||\hat{u} - \hat{u_h}||_0 \tag{16}$$

Thus

$$\begin{aligned} ||\hat{u} - \hat{u_h}||_0^2 &= ||\hat{e_h}||_0^2 \\ &= a(\hat{z}, \hat{e_h}) && \text{by 14} \\ &= a(\hat{z} - I_h \hat{z}, \hat{e_h}) && \text{by Galerkin orthogonality} \\ &\leq \alpha |\hat{z} - I_h \hat{z}|_1 |\hat{e_h}|_1 && \text{by continuity} \\ &\leq \alpha h |\hat{z}|_2 |\hat{e_h}|_1 && \text{by 15 et by interpolation inequality} \\ &\leq \alpha \cdot h |\hat{z}|_2 \cdot h^k ||\hat{\phi}||_\infty |C|_{k+1} && \text{by 13} \\ &\leq \alpha h^{k+1} ||\hat{u} - \hat{u_h}||_0 ||\hat{\phi}||_\infty |C|_{k+1} && \text{by 16} \end{aligned}$$

Finally

$$||\hat{u} - \hat{u_h}||_0 \leq \alpha h^{k+1} ||\hat{\phi}||_\infty |C|_{k+1}$$

Remark. *Note that*

$$||\hat{u} - \hat{u_h}||_0 = ||u + m - (u_h + m)||_0 = ||u - u_h||_0$$

4.4 Numerical results

As explained above, we wish to combine ϕ-FEM and FNO in order to predict the solution of the Poisson problem as accurately as possible. In this section, we present various results obtained using the 3 correction methods presented in the previous section (Section 4.2). It is important to note that, for practical purposes, almost all the following results obtained with ϕ-FEM will be compared with those obtained with the standard FEM method.
we will start by presenting the results obtained on an analytical solution (Section 4.4.1). we will consider here the "initial" solution $\tilde{\phi}$, which we will inject into the correction problems,

as the analytical solution of the problem. This first step simply enables us to check that, by supplying the exact solution directly to the correction solvers, they are indeed reduced to machine errors.
Next, in order to verify that correction solvers can improve accuracy when providing a solution close to the exact solution, we will consider the case of so-called "disturbed", firstly by applying an analytical perturbation (Section 4.4.2) and then by considering a ϕ-FEM solution as being the disturbed solution for which we don't know the perturbation (Section 4.4.3). This step will also provide us with a basis for further work, giving us an idea of what we can expect in terms of neural network output correction.
Finally, we will consider the case of neural networks with an FNO in Section 4.4.4 and then with other networks in Section 4.4.5 (a multi-perceptron network and a PINNs). The reasons for considering other neural networks will be explained in more details in these sections.
In this section, we defined by

$$||u_{ex} - u_{method}||^{(rel)}_{0,\Omega_h} = \frac{\int_{\Omega_h}(u_{ex} - u_{method})^2}{\int_{\Omega_h} u_{ex}^2}$$

the relative error between the exact solution u_{ex} and u_{method} a solution obtained by FEM or ϕ-FEM, a correction solver or the prediction of an neural network.
We can defined too

$$||u_{ex} - u_{method}||^{(abs)}_{0,\Omega_h} = \int_{\Omega_h}(u_{ex} - u_{method})^2$$

the absolute error.
In this section, we'll assume that $\tilde{\phi}$ is in $\mathbb{P}_k$ with $k = 10$.

4.4.1 Correction on exact solution

First, we will look at the correction of an exact solution. In other words, we consider

$$\tilde{\phi} = u_{ex}$$

This first step enables us to check that, by supplying an already exact solution to the various correctors under consideration, we again obtain an exact output solution (to the nearest machine error).
In all the following cases, we will consider $S = 0.5$, as well as $\varphi = 0$ in the case of the homogeneous problem and $\varphi = 1$ in the non-homogeneous case. We will vary f between 1 (low solution variability) and 4 (high solution variability) and we will choose to take 100 vertices in each direction $n_{vert} = 100$.

Remark. *We will consider here only the case of correction by addition (with and without an integration by parts) on $\Delta\tilde{\phi}$) as well as the case of correction by multiplication.*

- **Results on the Circle :**

  We consider here the Circle problem with the solution defined in Section 4.1.1.1.

Homogeneous case :

We consider here the homogeneous problem (i.e. with $\varphi = 0$) and seek to test the various correction methods with standard FEM and ϕ-FEM methods.

	FEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	2.10e-03	2.44e-10	1.29e-13	2.97e-13
f = 2	6.62e-03	1.53e-10	1.28e-13	2.80e-13
f = 3	1.41e-02	8.86e-11	1.27e-13	2.68e-13
f = 4	2.42e-02	9.52e-11	1.26e-13	2.61e-13

Figure 4.11: Relative errors obtained with different methods on the Circle with standard FEM.

	PhiFEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	8.05e-04	8.12e-11	3.32e-13	4.29e-13
f = 2	6.31e-03	6.50e-11	9.60e-12	1.12e-11
f = 3	2.04e-02	1.53e-10	1.45e-10	7.04e-11
f = 4	4.57e-02	8.78e-10	8.23e-10	1.04e-09

Figure 4.12: Relative errors obtained with different methods on the Circle with ϕ-FEM.

Non-homogeneous case :

We consider here the non-homogeneous problem (i.e. with $\varphi = 1$) and seek to test the various correction methods with standard FEM and ϕ-FEM methods.

	FEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	2.52e-03	2.48e-10	2.06e-14	2.10e-14
f = 2	1.05e-02	1.00e-10	1.66e-14	2.50e-14
f = 3	2.33e-02	7.67e-11	1.70e-14	2.60e-14
f = 4	4.07e-02	4.02e-11	1.56e-14	3.28e-14

Figure 4.13: Relative errors obtained with different methods on the Circle with standard FEM.

	PhiFEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	9.09e-05	2.97e-10	4.47e-13	1.68e-10
f = 2	3.97e-04	1.17e-10	4.55e-12	5.05e-11
f = 3	9.26e-04	8.05e-11	4.44e-11	2.89e-10
f = 4	1.66e-03	7.61e-10	7.65e-10	1.52e-09

Figure 4.14: Relative errors obtained with different methods on the Circle with ϕ-FEM.

- **Results on the Square :**

We consider here the Square problem with the solution defined in Section 4.1.2.1.

Homogeneous case :

We consider here the homogeneous problem (i.e. with $\varphi = 0$) and seek to test the various correction methods with standard FEM and ϕ-FEM methods.

	FEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	2.61e-03	5.16e-11	1.28e-13	2.48e-13
f = 2	1.04e-02	1.43e-11	1.27e-13	2.49e-13
f = 3	2.34e-02	8.07e-12	1.26e-13	2.49e-13
f = 4	4.12e-02	1.28e-11	1.24e-13	2.49e-13

Figure 4.15: Relative errors obtained with different methods on the Square with standard FEM.

	PhiFEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	1.84e-03	2.21e-11	2.21e-13	3.08e-13
f = 2	1.44e-02	8.80e-12	4.77e-12	4.80e-12
f = 3	3.69e-02	1.30e-10	1.29e-10	1.73e-10
f = 4	6.84e-02	1.27e-09	1.27e-09	2.02e-09

Figure 4.16: Relative errors obtained with different methods on the Square with ϕ-FEM.

Non-homogeneous case :

We consider here the non-homogeneous problem (i.e. with $\varphi = 1$) and seek to test the various correction methods with standard FEM and ϕ-FEM methods.

	FEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	2.30e-03	4.74e-11	1.10e-13	1.72e-13
f = 2	9.53e-03	1.79e-11	1.15e-13	2.16e-13
f = 3	2.19e-02	1.19e-11	1.17e-13	2.29e-13
f = 4	3.88e-02	8.32e-12	1.18e-13	2.35e-13

Figure 4.17: Relative errors obtained with different methods on the Square with standard FEM.

	PhiFEM	Corr_add	Corr_add_IPP	Corr_mult
f = 1	1.20e-04	1.42e-11	2.96e-13	2.02e-11
f = 2	5.74e-04	9.83e-12	4.45e-12	2.90e-11
f = 3	1.34e-03	1.16e-10	1.15e-10	1.78e-10
f = 4	2.39e-03	1.13e-09	1.14e-09	1.46e-09

Figure 4.18: Relative errors obtained with different methods on the Square with ϕ-FEM.

It would therefore seem that the various correction methods work in the different cases considered.

4.4.2 Correction on disturbed solution

Now, let's consider a deliberately disturbed solution. The purpose of this step is to check that the correction solvers also work with a solution that is very close to the real solution, but not exact. In this section, we will consider a manually disturbed solution, i.e. the exact solution to which we've added a small, analytically known perturbation.
As explained above, we begin by considering $\tilde{\phi}$ as a manually perturbed solution defined by

$$\tilde{\phi}(x,y) = u_{ex}(x,y) + \epsilon P(x,y)$$

where u_{ex} defines the exact solution to the problem, P the perturbation applied to it and ϵ is a real number that allows the amplitude of the perturbation to be easily increased or decreased. In this section, we'll assume that $\tilde{\phi}$ is in $\mathbb{P}_k$ with $k = 10$.

Remark. *Notice that by taking $\epsilon = 0$, we return to the case of correction on an exact solution presented in Section 4.4.1.*

In our case, we will choose to consider P as being of the same form as our exact solution (defined with different parameters), but we could very well consider a completely different perturbation.

Remark. *Note that the form of the perturbation has a huge influence on the accuracy of the solvers, and that the difficulty lies in the following cases where its expression is not explicitly known (as in the case of ϕ-FEM in Section 4.4.3 or FNO in Section 4.4.4).*

In the case of Circle geometry where we consider the problem 4.1.1.1, the perturbation will be defined by

$$P(x,y) = S_p \times \sin\left(8\pi f_p\left((x-0.5)^2 + (y-0.5)^2\right) + \varphi_p\right)$$

where $S_p \in [0,1]$ is the amplitude of the signal, $f_p \in \mathbb{N}$ can be associated with the "frequency" of the signal and $\varphi_p \in [0,1]$ the phase at the origin.
In the case of Square geometry where we consider the problem 4.1.2.1, the perturbation will be defined by

$$P(x,y) = S_p \times \sin\left(2\pi f_p x + \varphi_p\right) \times \sin\left(2\pi f_p y + \varphi_p\right)$$

where $S_p \in [0,1]$ is the amplitude of the signal, $f_p \in \mathbb{N}$ can be associated with the "frequency" of the signal and $\varphi_p \in [0,1]$ the phase at the origin.

Remark. *Note that for the boundary conditions of the solution to be satisfied, i.e. for $\tilde{\phi} = u_{ex}$ on Γ, it is essential that $P = 0$ on Γ. In the case of both circle and square, we will then take $\varphi_p = 0$.*

Recall the relative errors obtained by standard FEM and ϕ-FEM on the circle and on the square for frequencies $f \in \{1,2,3,4\}$ for the homogeneous (Figure 4.19) and non-homogeneous problem (Figure 4.20).

		f = 1	f = 2	f = 3	f = 4
circle	fem	2.10e-03	6.62e-03	1.41e-02	2.42e-02
	phifem	8.05e-04	6.31e-03	2.04e-02	4.57e-02
square	fem	2.61e-03	1.04e-02	2.34e-02	4.12e-02
	phifem	1.84e-03	1.44e-02	3.69e-02	6.84e-02

Figure 4.19: Table summarizing the errors obtained by standard FEM and ϕ-FEM on the circle and the square (homogeneous case).

		f = 1	f = 2	f = 3	f = 4
circle	fem	2.52e-03	1.05e-02	2.33e-02	4.07e-02
	phifem	9.09e-05	3.97e-04	9.26e-04	1.66e-03
square	fem	2.30e-03	9.53e-03	2.19e-02	3.88e-02
	phifem	1.20e-04	5.74e-04	1.34e-03	2.39e-03

Figure 4.20: Table summarizing the errors obtained by standard FEM and ϕ-FEM on the circle and the square (non-homogeneous case).

Remark. *In the previous results, we use the direct method to impose the boundary conditions with ϕ-FEM.*

4.4.2.1 Results with different ϵ

The aim of this section is to test correction by addition (without integration by parts) and correction by multiplication by varying the amplitude of the perturbation (in other words, by varying ϵ).
We will try to separate the cases according to the frequencies considered. In other words, for $f, f_p \in \{1,2,3,4\}$, we're interested in the following three cases. The first is the case where the solution frequency is greater than the perturbation frequency ($f > f_p$), i.e. a highly variable solution and a less variable perturbation. The second is where the solution and perturbation frequencies are equal ($f = f_p$), i.e. the solution and perturbation have the same variability. The last category covers cases where the perturbation is "nastier" than the solution, i.e. it has a higher frequency than the solution ($f < f_p$).
In this section, we will consider for the circle the solution defined in Section 4.1.1.1 and for the square the solution defined in Section 4.1.2.1.
Results with standard FEM :
First we will consider the standard FEM method on the homogeneous case (i.e. with $\varphi = 0$) and then on the non-homogeneous case (i.e. with $\varphi = 1$).

- **Results on the homogeneous case :**

  First, we consider the correction by adding (without integration by parts) on the Circle problem (Figure 4.21) and on the Square problem (Figure 4.22).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	1.53e-10
	f = 3 ; fp = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	8.86e-11
	f = 3 ; fp = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	8.86e-11
	f = 4 ; fp = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	9.52e-11
	f = 4 ; fp = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	9.52e-11
	f = 4 ; fp = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	9.52e-11
f = fp	f = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	2.44e-10
	f = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	1.53e-10
	f = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	8.86e-11
	f = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	9.52e-11
f < fp	f = 1 ; fp = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	2.44e-10
	f = 1 ; fp = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	2.44e-10
	f = 1 ; fp = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	2.44e-10
	f = 2 ; fp = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	1.53e-10
	f = 2 ; fp = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	1.53e-10
	f = 3 ; fp = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	8.86e-11

Figure 4.21: Correction by adding on the Circle with standard FEM in the homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	1.28e-11
	f = 3 ; fp = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	8.33e-12
	f = 3 ; fp = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	8.33e-12
	f = 4 ; fp = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	1.08e-11
	f = 4 ; fp = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	1.08e-11
	f = 4 ; fp = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	1.08e-11
f = fp	f = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	4.16e-11
	f = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	1.28e-11
	f = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	8.33e-12
	f = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	1.08e-11
f < fp	f = 1 ; fp = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	4.16e-11
	f = 1 ; fp = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	4.16e-11
	f = 1 ; fp = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	4.16e-11
	f = 2 ; fp = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	1.28e-11
	f = 2 ; fp = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	1.28e-11
	f = 3 ; fp = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	8.33e-12

Figure 4.22: Correction by adding on the Square with standard FEM in the homogeneous case.

Then, we consider the correction by multiplying on the Circle problem (Figure 4.23) and on the Square problem (Figure 4.24).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	9.76e-01	1.04e-01	1.04e-02	1.04e-03	1.04e-04	2.80e-13
	f = 3 ; fp = 1	1.03e+00	1.03e-01	1.02e-02	1.02e-03	1.02e-04	2.68e-13
	f = 3 ; fp = 2	7.60e-01	8.62e-02	8.64e-03	8.65e-04	8.65e-05	2.68e-13
	f = 4 ; fp = 1	1.00e+00	1.01e-01	1.01e-02	1.01e-03	1.01e-04	2.61e-13
	f = 4 ; fp = 2	8.60e-01	9.49e-02	9.46e-03	9.46e-04	9.46e-05	2.61e-13
	f = 4 ; fp = 3	6.92e-01	7.70e-02	7.73e-03	7.73e-04	7.73e-05	2.61e-13
f = fp	f = 1	5.23e-02	5.23e-03	5.23e-04	5.23e-05	5.23e-06	2.97e-13
	f = 2	5.32e-02	5.32e-03	5.32e-04	5.32e-05	5.32e-06	2.80e-13
	f = 3	6.33e-02	6.33e-03	6.33e-04	6.33e-05	6.33e-06	2.68e-13
	f = 4	7.19e-02	7.19e-03	7.19e-04	7.19e-05	7.19e-06	2.61e-13
f < fp	f = 1 ; fp = 2	9.80e-01	1.08e-02	1.07e-03	1.06e-04	1.06e-05	2.97e-13
	f = 1 ; fp = 3	1.01e+00	1.56e-02	1.64e-03	1.64e-04	1.64e-05	2.97e-13
	f = 1 ; fp = 4	1.01e+00	2.55e-02	2.20e-03	2.17e-04	2.17e-05	2.97e-13
	f = 2 ; fp = 3	7.63e-01	9.93e-02	1.01e-02	1.01e-03	1.01e-04	2.80e-13
	f = 2 ; fp = 4	8.38e-01	1.08e-02	1.10e-03	1.10e-04	1.10e-05	2.80e-13
	f = 3 ; fp = 4	6.95e-01	8.24e-02	8.40e-03	8.42e-04	8.42e-05	2.68e-13

Figure 4.23: Correction by multiplying on the Circle with standard FEM in the homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	6.45e-01	8.61e-02	8.68e-03	8.68e-04	8.68e-05	2.66e-14
	f = 3 ; fp = 1	7.87e-01	9.35e-02	9.39e-03	9.40e-04	9.40e-05	2.67e-14
	f = 3 ; fp = 2	5.74e-01	7.98e-02	8.04e-03	8.04e-04	8.04e-05	2.67e-14
	f = 4 ; fp = 1	8.48e-01	9.66e-02	9.67e-03	9.67e-04	9.67e-05	3.71e-14
	f = 4 ; fp = 2	6.61e-01	8.75e-02	8.82e-03	8.82e-04	8.82e-05	3.71e-14
	f = 4 ; fp = 3	5.60e-01	7.75e-02	7.81e-03	7.81e-04	7.81e-05	3.71e-14
f = fp	f = 1	2.78e-02	2.78e-03	2.78e-04	2.78e-05	2.78e-06	2.56e-14
	f = 2	3.72e-02	3.72e-03	3.72e-04	3.72e-05	3.72e-06	2.66e-14
	f = 3	4.66e-02	4.66e-03	4.66e-04	4.66e-05	4.66e-06	2.67e-14
	f = 4	5.42e-02	5.42e-03	5.42e-04	5.42e-05	5.42e-06	3.71e-14
f < fp	f = 1 ; fp = 2	6.37e-01	4.01e-03	3.98e-04	3.98e-05	3.98e-06	2.56e-14
	f = 1 ; fp = 3	7.82e-01	5.70e-03	5.53e-04	5.55e-05	5.55e-06	2.56e-14
	f = 1 ; fp = 4	8.44e-01	2.89e-02	7.42e-04	7.39e-05	7.39e-06	2.56e-14
	f = 2 ; fp = 3	5.68e-01	6.57e-02	6.77e-03	6.78e-04	6.78e-05	2.66e-14
	f = 2 ; fp = 4	6.54e-01	6.58e-03	6.38e-04	6.38e-05	6.38e-06	2.66e-14
	f = 3 ; fp = 4	5.56e-01	6.82e-02	6.94e-03	6.94e-04	6.94e-05	2.67e-14

Figure 4.24: Correction by multiplying on the Square with standard FEM in the homogeneous case.

Observation : We can make a few comments on the results obtained:

- First of all, it would therefore seem that, overall, the smaller the perturbation applied (i.e. the smaller the ϵ), the more efficient the addition and multiplication correction solvers are in terms of accuracy.

- Then, it would appear that, as with the standard FEM and ϕ-FEM solvers without correction, the more the solution varies (i.e. the larger f), the greater the error. This is a fairly intuitive result, since the more the solution varies, the more points are needed to approximate it.
- It would also seem that for $\epsilon = 1$ (i.e. a large perturbation), the ϵ parameter has a greater impact on the multiplicative corrector than on the additive corrector. We explained earlier the benefits of elevating the problem, which could be beneficial here. Results on elevation will be presented in the Section 4.4.2.2.
- In view of the results obtained here, it would also appear that, overall, correction by addition is more effective than correction by multiplication. Moreover, correction by addition has more advantages than correction by multiplication. In particular, if the solution cancels out on the domain, correction by multiplication will require elevating the problem sufficiently so that it no longer cancels out, unlike correction by addition.
- There is one final and rather important point to make. In fact, if we take a closer look at the results, we can see that in the case of correction by adding, the errors only seem to depend on the frequency of the perturbation and not on that of the solution (at a fixed ϵ). This is a result that has been explained theoretically in the case of correction by multiplication on a elevated problem in the Section 4.3.1 (for m large, similar to correction by addition as explained above). Thus, as we have shown (in Section 4.3.2) that for m large, the error of correction by multiplication on a elevated problem converges to the error of correction by addition, we recover this result on correction by addition.

- **Results on the non-homogeneous case :**

  First, we consider the correction by adding (without IPP) on the Circle problem (Figure 4.25) and on the Square problem (Figure 4.26).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	9.95e-11
	f = 3 ; fp = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	7.08e-11
	f = 3 ; fp = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	7.08e-11
	f = 4 ; fp = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	6.02e-11
	f = 4 ; fp = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	6.02e-11
	f = 4 ; fp = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	6.02e-11
f = fp	f = 1	2.10e-03	2.10e-04	2.10e-05	2.10e-06	2.10e-07	2.90e-10
	f = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	9.95e-11
	f = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	7.08e-11
	f = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	6.02e-11
f < fp	f = 1 ; fp = 2	6.62e-03	6.62e-04	6.62e-05	6.62e-06	6.62e-07	2.90e-10
	f = 1 ; fp = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	2.90e-10
	f = 1 ; fp = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	2.90e-10
	f = 2 ; fp = 3	1.41e-02	1.41e-03	1.41e-04	1.41e-05	1.41e-06	9.95e-11
	f = 2 ; fp = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	9.95e-11
	f = 3 ; fp = 4	2.42e-02	2.42e-03	2.42e-04	2.42e-05	2.42e-06	7.08e-11

Figure 4.25: Correction by adding on the Circle with standard FEM in the non-homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	1.79e-11
	f = 3 ; fp = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	1.19e-11
	f = 3 ; fp = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	1.19e-11
	f = 4 ; fp = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	8.32e-12
	f = 4 ; fp = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	8.32e-12
	f = 4 ; fp = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	8.32e-12
f = fp	f = 1	2.61e-03	2.61e-04	2.61e-05	2.61e-06	2.61e-07	4.74e-11
	f = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	1.79e-11
	f = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	1.19e-11
	f = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	8.32e-12
f < fp	f = 1 ; fp = 2	1.04e-02	1.04e-03	1.04e-04	1.04e-05	1.04e-06	4.74e-11
	f = 1 ; fp = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	4.74e-11
	f = 1 ; fp = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	4.74e-11
	f = 2 ; fp = 3	2.34e-02	2.34e-03	2.34e-04	2.34e-05	2.34e-06	1.79e-11
	f = 2 ; fp = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	1.79e-11
	f = 3 ; fp = 4	4.12e-02	4.12e-03	4.12e-04	4.12e-05	4.12e-06	1.19e-11

Figure 4.26: Correction by adding on the Square with standard FEM in the non-homogeneous case.

Then, we consider the correction by multiplying on the Circle problem (Figure 4.27) and on the Square problem (Figure 4.28).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.20e+00	1.03e-01	1.00e-02	1.00e-03	1.00e-04	2.42e-13
	f = 3 ; fp = 1	1.02e+00	9.96e-02	9.91e-03	9.91e-04	9.91e-05	2.47e-13
	f = 3 ; fp = 2	9.91e-01	9.26e-02	9.03e-03	9.00e-04	9.00e-05	2.47e-13
	f = 4 ; fp = 1	9.70e-01	9.92e-02	9.92e-03	9.92e-04	9.92e-05	2.46e-13
	f = 4 ; fp = 2	9.67e-01	9.37e-02	9.30e-03	9.29e-04	9.29e-05	2.46e-13
	f = 4 ; fp = 3	8.97e-01	8.55e-02	8.35e-03	8.33e-04	8.33e-05	2.46e-13
f = fp	f = 1	6.61e-01	1.02e-01	1.06e-02	1.07e-03	1.07e-04	2.17e-13
	f = 2	5.49e-01	8.74e-02	9.13e-03	9.17e-04	9.17e-05	2.42e-13
	f = 3	5.15e-01	8.22e-02	8.60e-03	8.64e-04	8.64e-05	2.47e-13
	f = 4	4.95e-01	7.96e-02	8.34e-03	8.37e-04	8.38e-05	2.46e-13
f < fp	f = 1 ; fp = 2	4.42e-01	1.04e-01	1.14e-02	1.15e-03	1.15e-04	2.17e-13
	f = 1 ; fp = 3	8.20e-01	2.53e-02	1.84e-03	1.79e-04	1.79e-05	2.17e-13
	f = 1 ; fp = 4	8.56e-01	7.39e-02	9.34e-03	9.56e-04	9.58e-05	2.17e-13
	f = 2 ; fp = 3	4.31e-01	7.23e-02	7.27e-03	7.27e-04	7.27e-05	2.42e-13
	f = 2 ; fp = 4	7.27e-01	1.21e-01	1.23e-02	1.23e-03	1.23e-04	2.42e-13
	f = 3 ; fp = 4	4.40e-01	6.91e-02	7.13e-03	7.15e-04	7.16e-05	2.47e-13

Figure 4.27: Correction by multiplying on the Circle with standard FEM in the non-homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	6.42e-01	8.64e-02	8.68e-03	8.67e-04	8.67e-05	2.16e-13
	f = 3 ; fp = 1	7.86e-01	9.36e-02	9.39e-03	9.40e-04	9.40e-05	2.29e-13
	f = 3 ; fp = 2	5.67e-01	7.97e-02	8.02e-03	8.02e-04	8.02e-05	2.29e-13
	f = 4 ; fp = 1	8.44e-01	9.66e-02	9.67e-03	9.67e-04	9.67e-05	2.35e-13
	f = 4 ; fp = 2	6.56e-01	8.78e-02	8.81e-03	8.81e-04	8.81e-05	2.35e-13
	f = 4 ; fp = 3	5.60e-01	7.74e-02	7.79e-03	7.79e-04	7.79e-05	2.35e-13
f = fp	f = 1	3.76e-01	7.12e-02	7.66e-03	7.70e-04	7.70e-05	1.72e-13
	f = 2	3.95e-01	7.31e-02	7.81e-03	7.85e-04	7.86e-05	2.16e-13
	f = 3	4.09e-01	7.47e-02	7.98e-03	8.02e-04	8.02e-05	2.29e-13
	f = 4	4.22e-01	7.64e-02	8.13e-03	8.17e-04	8.18e-05	2.35e-13
f < fp	f = 1 ; fp = 2	5.23e-01	7.12e-02	7.39e-03	7.39e-04	7.39e-05	1.72e-13
	f = 1 ; fp = 3	6.62e-01	9.68e-03	9.73e-04	9.96e-05	9.99e-06	1.72e-13
	f = 1 ; fp = 4	7.18e-01	5.27e-02	4.69e-03	4.69e-04	4.69e-05	1.72e-13
	f = 2 ; fp = 3	5.35e-01	6.31e-02	6.51e-03	6.52e-04	6.52e-05	2.16e-13
	f = 2 ; fp = 4	5.88e-01	7.69e-02	7.98e-03	7.99e-04	7.99e-05	2.16e-13
	f = 3 ; fp = 4	5.42e-01	6.72e-02	6.81e-03	6.82e-04	6.82e-05	2.29e-13

Figure 4.28: Correction by multiplying on the Square with standard FEM in the non-homogeneous case.

Observation : In view of the results obtained, it would appear that the conclusions are the same as for the homogeneous case.

Remark. *Considering $\tilde{\phi}$ in $\mathbb{P}_2$ and the homogeneous problem, we'll apply the correction by adding on the Circle problem (Figure 4.29) and on the Square problem (Figure 4.30).*

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	3.33e-03	3.57e-03	3.69e-03	3.70e-03	3.70e-03	3.70e-03
	f = 3 ; fp = 1	9.09e-03	8.29e-03	8.25e-03	8.24e-03	8.24e-03	8.24e-03
	f = 3 ; fp = 2	8.52e-03	7.73e-03	8.19e-03	8.24e-03	8.24e-03	8.24e-03
	f = 4 ; fp = 1	1.49e-02	1.49e-02	1.49e-02	1.49e-02	1.49e-02	1.49e-02
	f = 4 ; fp = 2	1.90e-02	1.51e-02	1.49e-02	1.49e-02	1.49e-02	1.49e-02
	f = 4 ; fp = 3	1.80e-02	1.38e-02	1.48e-02	1.49e-02	1.49e-02	1.49e-02
f = fp	f = 1	3.43e-03	1.09e-03	8.91e-04	8.72e-04	8.70e-04	8.70e-04
	f = 2	1.33e-02	4.61e-03	3.79e-03	3.71e-03	3.70e-03	3.70e-03
	f = 3	2.94e-02	1.03e-02	8.44e-03	8.26e-03	8.24e-03	8.24e-03
	f = 4	5.20e-02	1.85e-02	1.53e-02	1.50e-02	1.49e-02	1.49e-02
f < fp	f = 1 ; fp = 2	9.37e-03	8.88e-04	8.20e-04	8.64e-04	8.69e-04	8.70e-04
	f = 1 ; fp = 3	2.17e-02	2.42e-03	9.21e-04	8.73e-04	8.70e-04	8.70e-04
	f = 1 ; fp = 4	3.77e-02	3.84e-03	9.34e-04	8.69e-04	8.69e-04	8.70e-04
	f = 2 ; fp = 3	1.97e-02	3.07e-03	3.58e-03	3.69e-03	3.70e-03	3.70e-03
	f = 2 ; fp = 4	3.84e-02	5.60e-03	3.77e-03	3.70e-03	3.70e-03	3.70e-03
	f = 3 ; fp = 4	3.37e-02	6.81e-03	8.03e-03	8.22e-03	8.24e-03	8.24e-03

Figure 4.29: Correction by adding on the Circle with standard FEM in the homogeneous case ($\tilde{\phi}$ in $\mathbb{P}_2$).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	7.68e-03	6.53e-03	6.52e-03	6.52e-03	6.52e-03	6.52e-03
	f = 3 ; fp = 1	1.51e-02	1.46e-02	1.45e-02	1.46e-02	1.46e-02	1.46e-02
	f = 3 ; fp = 2	2.21e-02	1.47e-02	1.46e-02	1.46e-02	1.46e-02	1.46e-02
	f = 4 ; fp = 1	2.60e-02	2.58e-02	2.58e-02	2.58e-02	2.58e-02	2.58e-02
	f = 4 ; fp = 2	3.02e-02	2.58e-02	2.58e-02	2.58e-02	2.58e-02	2.58e-02
	f = 4 ; fp = 3	4.47e-02	2.61e-02	2.58e-02	2.58e-02	2.58e-02	2.58e-02
f = fp	f = 1	5.72e-03	2.04e-03	1.69e-03	1.65e-03	1.65e-03	1.65e-03
	f = 2	2.27e-02	8.10e-03	6.68e-03	6.54e-03	6.52e-03	6.52e-03
	f = 3	5.06e-02	1.81e-02	1.49e-02	1.46e-02	1.46e-02	1.46e-02
	f = 4	8.92e-02	3.20e-02	2.64e-02	2.59e-02	2.58e-02	2.58e-02
f < fp	f = 1 ; fp = 2	1.65e-02	2.32e-03	1.65e-03	1.65e-03	1.65e-03	1.65e-03
	f = 1 ; fp = 3	3.67e-02	4.01e-03	1.68e-03	1.65e-03	1.65e-03	1.65e-03
	f = 1 ; fp = 4	6.45e-02	6.64e-03	1.76e-03	1.65e-03	1.65e-03	1.65e-03
	f = 2 ; fp = 3	3.73e-02	7.51e-03	6.54e-03	6.52e-03	6.52e-03	6.52e-03
	f = 2 ; fp = 4	6.47e-02	9.10e-03	6.54e-03	6.52e-03	6.52e-03	6.52e-03
	f = 3 ; fp = 4	6.61e-02	1.59e-02	1.46e-02	1.46e-02	1.46e-02	1.46e-02

Figure 4.30: Correction by adding on the Square with standard FEM in the homogeneous case ($\tilde{\phi}$ in $\mathbb{P}_2$).

And by considering the correction by multiplying on the Circle problem (Figure 4.31) and on the Square problem (Figure 4.32).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	9.76e-01	1.04e-01	1.04e-02	1.04e-03	1.07e-04	1.37e-05
	f = 3 ; fp = 1	1.03e+00	1.03e-01	1.02e-02	1.03e-03	1.22e-04	5.73e-05
	f = 3 ; fp = 2	7.60e-01	8.62e-02	8.65e-03	8.72e-04	1.08e-04	5.73e-05
	f = 4 ; fp = 1	1.00e+00	1.01e-01	1.01e-02	1.02e-03	1.91e-04	1.64e-04
	f = 4 ; fp = 2	8.60e-01	9.49e-02	9.47e-03	9.70e-04	1.94e-04	1.64e-04
	f = 4 ; fp = 3	6.92e-01	7.71e-02	7.75e-03	8.01e-04	1.86e-04	1.64e-04
f = fp	f = 1	5.23e-02	5.23e-03	5.23e-04	5.26e-05	5.74e-06	1.44e-06
	f = 2	5.33e-02	5.33e-03	5.34e-04	5.64e-05	1.53e-05	1.37e-05
	f = 3	6.34e-02	6.36e-03	6.48e-04	9.37e-05	5.89e-05	5.73e-05
	f = 4	7.23e-02	7.26e-03	7.79e-04	1.95e-04	1.66e-04	1.64e-04
f < fp	f = 1 ; fp = 2	9.80e-01	1.08e-02	1.07e-03	1.06e-04	1.04e-05	1.44e-06
	f = 1 ; fp = 3	1.01e+00	1.57e-02	1.64e-03	1.65e-04	1.68e-05	1.44e-06
	f = 1 ; fp = 4	1.01e+00	2.55e-02	2.20e-03	2.17e-04	2.14e-05	1.44e-06
	f = 2 ; fp = 3	7.63e-01	9.93e-02	1.01e-02	1.01e-03	9.93e-05	1.37e-05
	f = 2 ; fp = 4	8.38e-01	1.08e-02	1.10e-03	1.13e-04	1.85e-05	1.37e-05
	f = 3 ; fp = 4	6.95e-01	8.24e-02	8.40e-03	8.37e-04	9.59e-05	5.73e-05

Figure 4.31: Correction by multiplying on the Circle with standard FEM in the homogeneous case ($\tilde{\phi}$ in $\mathbb{P}_2$).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	6.45e-01	8.61e-02	8.68e-03	8.68e-04	9.21e-05	3.27e-05
	f = 3 ; fp = 1	7.87e-01	9.35e-02	9.39e-03	9.46e-04	1.59e-04	1.30e-04
	f = 3 ; fp = 2	5.74e-01	7.99e-02	8.04e-03	8.13e-04	1.52e-04	1.30e-04
	f = 4 ; fp = 1	8.48e-01	9.66e-02	9.68e-03	1.02e-03	3.50e-04	3.38e-04
	f = 4 ; fp = 2	6.61e-01	8.75e-02	8.82e-03	9.35e-04	3.47e-04	3.38e-04
	f = 4 ; fp = 3	5.60e-01	7.75e-02	7.82e-03	8.46e-04	3.46e-04	3.38e-04
f = fp	f = 1	2.78e-02	2.78e-03	2.80e-04	3.03e-05	6.01e-06	3.89e-06
	f = 2	3.73e-02	3.73e-03	3.83e-04	5.64e-05	3.40e-05	3.27e-05
	f = 3	4.67e-02	4.69e-03	5.03e-04	1.44e-04	1.31e-04	1.30e-04
	f = 4	5.44e-02	5.48e-03	6.68e-04	3.48e-04	3.39e-04	3.38e-04
f < fp	f = 1 ; fp = 2	6.37e-01	4.01e-03	3.98e-04	4.02e-05	5.71e-06	3.89e-06
	f = 1 ; fp = 3	7.82e-01	5.70e-03	5.54e-04	5.60e-05	7.07e-06	3.89e-06
	f = 1 ; fp = 4	8.44e-01	2.88e-02	7.42e-04	7.39e-05	8.28e-06	3.89e-06
	f = 2 ; fp = 3	5.68e-01	6.57e-02	6.78e-03	6.79e-04	7.56e-05	3.27e-05
	f = 2 ; fp = 4	6.54e-01	6.58e-03	6.39e-04	7.18e-05	3.34e-05	3.27e-05
	f = 3 ; fp = 4	5.56e-01	6.83e-02	6.94e-03	7.06e-04	1.47e-04	1.30e-04

Figure 4.32: Correction by multiplying on the Square with standard FEM in the homogeneous case ($\tilde{\phi}$ in $\mathbb{P}_2$).

Observation : *In view of the results obtained here, it seems that considering ϕ only in a space of degree 2 is not sufficient to considerably improve the errors obtained by standard methods, even when considering an exact solution.*

Results with ϕ-FEM :

Then we will consider the ϕ-FEM method on the homogeneous case (i.e. with $\varphi = 0$) and then on the non-homogeneous case (i.e. with $\varphi = 1$).

- **Results on the homogeneous case :**

  First, we consider the correction by adding (without integration by parts) on the Circle problem (Figure 4.33) and on the Square problem (Figure 4.34).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	8.14e-04	8.14e-05	8.14e-06	8.14e-07	8.15e-08	6.50e-11
	f = 3 ; fp = 1	8.13e-04	8.13e-05	8.13e-06	8.13e-07	8.13e-08	1.53e-10
	f = 3 ; fp = 2	6.52e-03	6.52e-04	6.52e-05	6.52e-06	6.52e-07	1.53e-10
	f = 4 ; fp = 1	8.13e-04	8.13e-05	8.13e-06	8.13e-07	8.17e-08	8.78e-10
	f = 4 ; fp = 2	6.51e-03	6.51e-04	6.51e-05	6.51e-06	6.50e-07	8.78e-10
	f = 4 ; fp = 3	2.15e-02	2.15e-03	2.15e-04	2.15e-05	2.15e-06	8.78e-10
f = fp	f = 1	8.14e-04	8.14e-05	8.14e-06	8.15e-07	8.15e-08	8.12e-11
	f = 2	6.52e-03	6.52e-04	6.52e-05	6.52e-06	6.52e-07	6.50e-11
	f = 3	2.15e-02	2.15e-03	2.15e-04	2.15e-05	2.15e-06	1.53e-10
	f = 4	4.89e-02	4.89e-03	4.89e-04	4.89e-05	4.89e-06	8.78e-10
f < fp	f = 1 ; fp = 2	6.52e-03	6.52e-04	6.52e-05	6.52e-06	6.52e-07	8.12e-11
	f = 1 ; fp = 3	2.16e-02	2.16e-03	2.16e-04	2.16e-05	2.16e-06	8.12e-11
	f = 1 ; fp = 4	4.90e-02	4.90e-03	4.90e-04	4.90e-05	4.90e-06	8.12e-11
	f = 2 ; fp = 3	2.15e-02	2.15e-03	2.15e-04	2.15e-05	2.15e-06	6.50e-11
	f = 2 ; fp = 4	4.90e-02	4.90e-03	4.90e-04	4.90e-05	4.90e-06	6.50e-11
	f = 3 ; fp = 4	4.90e-02	4.90e-03	4.90e-04	4.90e-05	4.90e-06	1.53e-10

Figure 4.33: Correction by adding on the Circle with ϕ-FEM in the homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.86e-03	1.86e-04	1.86e-05	1.86e-06	1.86e-07	1.33e-11
	f = 3 ; fp = 1	1.86e-03	1.86e-04	1.86e-05	1.86e-06	1.86e-07	1.24e-10
	f = 3 ; fp = 2	1.50e-02	1.50e-03	1.50e-04	1.50e-05	1.50e-06	1.24e-10
	f = 4 ; fp = 1	1.86e-03	1.86e-04	1.86e-05	1.86e-06	1.86e-07	1.27e-09
	f = 4 ; fp = 2	1.50e-02	1.50e-03	1.50e-04	1.50e-05	1.50e-06	1.27e-09
	f = 4 ; fp = 3	4.09e-02	4.09e-03	4.09e-04	4.09e-05	4.09e-06	1.27e-09
f = fp	f = 1	1.86e-03	1.86e-04	1.86e-05	1.86e-06	1.86e-07	5.43e-11
	f = 2	1.51e-02	1.51e-03	1.51e-04	1.51e-05	1.51e-06	1.33e-11
	f = 3	4.10e-02	4.10e-03	4.10e-04	4.10e-05	4.10e-06	1.24e-10
	f = 4	8.08e-02	8.08e-03	8.08e-04	8.08e-05	8.08e-06	1.27e-09
f < fp	f = 1 ; fp = 2	1.51e-02	1.51e-03	1.51e-04	1.51e-05	1.51e-06	5.43e-11
	f = 1 ; fp = 3	4.10e-02	4.10e-03	4.10e-04	4.10e-05	4.10e-06	5.43e-11
	f = 1 ; fp = 4	8.11e-02	8.11e-03	8.11e-04	8.11e-05	8.11e-06	5.43e-11
	f = 2 ; fp = 3	4.10e-02	4.10e-03	4.10e-04	4.10e-05	4.10e-06	1.33e-11
	f = 2 ; fp = 4	8.10e-02	8.10e-03	8.10e-04	8.10e-05	8.10e-06	1.33e-11
	f = 3 ; fp = 4	8.09e-02	8.09e-03	8.09e-04	8.09e-05	8.09e-06	1.24e-10

Figure 4.34: Correction by adding on the Square with ϕ-FEM in the homogeneous case.

Then, we consider the correction by multiplying on the Circle problem (Figure 4.35) and on the Square problem (Figure 4.36).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	9.73e-01	1.04e-01	1.03e-02	1.03e-03	1.03e-04	1.12e-11
	f = 3 ; fp = 1	1.02e+00	1.03e-01	1.02e-02	1.02e-03	1.02e-04	7.04e-11
	f = 3 ; fp = 2	7.61e-01	8.59e-02	8.60e-03	8.60e-04	8.60e-05	7.04e-11
	f = 4 ; fp = 1	9.96e-01	1.01e-01	1.01e-02	1.01e-03	1.01e-04	1.04e-09
	f = 4 ; fp = 2	8.58e-01	9.48e-02	9.45e-03	9.45e-04	9.45e-05	1.04e-09
	f = 4 ; fp = 3	6.96e-01	7.69e-02	7.72e-03	7.72e-04	7.72e-05	1.04e-09
f = fp	f = 1	4.29e-13	4.24e-13	4.26e-13	4.29e-13	4.27e-13	4.29e-13
	f = 2	1.12e-11	1.12e-11	1.12e-11	1.12e-11	1.12e-11	1.12e-11
	f = 3	7.04e-11	7.04e-11	7.04e-11	7.04e-11	7.04e-11	7.04e-11
	f = 4	1.04e-09	1.04e-09	1.04e-09	1.04e-09	1.04e-09	1.04e-09
f < fp	f = 1 ; fp = 2	9.74e-01	4.66e-04	7.55e-05	7.76e-06	7.78e-07	4.29e-13
	f = 1 ; fp = 3	1.02e+00	6.28e-03	3.56e-04	3.14e-05	3.10e-06	4.29e-13
	f = 1 ; fp = 4	9.98e-01	5.23e-02	5.75e-04	8.60e-05	8.88e-06	4.29e-13
	f = 2 ; fp = 3	7.61e-01	1.02e-01	1.04e-02	1.04e-03	1.04e-04	1.12e-11
	f = 2 ; fp = 4	8.59e-01	1.07e-02	9.31e-04	9.07e-05	9.04e-06	1.12e-11
	f = 3 ; fp = 4	6.96e-01	8.63e-02	8.72e-03	8.73e-04	8.73e-05	7.04e-11

Figure 4.35: Correction by multiplying on the Circle with ϕ-FEM in the homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	6.45e-01	8.60e-02	8.71e-03	8.72e-04	8.72e-05	4.38e-12
	f = 3 ; fp = 1	7.89e-01	9.34e-02	9.41e-03	9.41e-04	9.41e-05	1.72e-10
	f = 3 ; fp = 2	5.79e-01	8.01e-02	8.11e-03	8.11e-04	8.11e-05	1.72e-10
	f = 4 ; fp = 1	8.49e-01	9.64e-02	9.67e-03	9.67e-04	9.67e-05	2.02e-09
	f = 4 ; fp = 2	6.68e-01	8.78e-02	8.86e-03	8.86e-04	8.86e-05	2.02e-09
	f = 4 ; fp = 3	5.70e-01	7.83e-02	7.94e-03	7.94e-04	7.94e-05	2.02e-09
f = fp	f = 1	3.37e-13	3.38e-13	3.39e-13	3.39e-13	3.38e-13	3.37e-13
	f = 2	4.38e-12	4.38e-12	4.38e-12	4.38e-12	4.39e-12	4.38e-12
	f = 3	1.72e-10	1.72e-10	1.72e-10	1.72e-10	1.72e-10	1.72e-10
	f = 4	2.02e-09	2.02e-09	2.02e-09	2.02e-09	2.02e-09	2.02e-09
f < fp	f = 1 ; fp = 2	6.46e-01	9.19e-04	7.68e-05	7.67e-06	7.67e-07	3.37e-13
	f = 1 ; fp = 3	7.90e-01	4.83e-03	2.93e-04	2.93e-05	2.93e-06	3.37e-13
	f = 1 ; fp = 4	8.51e-01	3.99e-02	6.66e-04	6.38e-05	6.38e-06	3.37e-13
	f = 2 ; fp = 3	5.80e-01	6.72e-02	7.07e-03	7.08e-04	7.08e-05	4.38e-12
	f = 2 ; fp = 4	6.70e-01	4.10e-03	3.40e-04	3.39e-05	3.39e-06	4.38e-12
	f = 3 ; fp = 4	5.70e-01	7.12e-02	7.30e-03	7.30e-04	7.30e-05	1.72e-10

Figure 4.36: Correction by multiplying on the Square with ϕ-FEM in the homogeneous case.

Observation : An interesting result can also be observed. Indeed, it seems that in the case where $f = f_p$, the multiplication correction with ϕ-FEM seems to approach the solution almost perfectly for all ϵ considered. In fact, in the homogeneous case, for $f = f_p$ the perturbation is identical to the solution (i.e. $P = u_{ex}$) and so the solution injected into the correction solvers is of the form

$$\tilde{\phi} = u_{ex} + \epsilon P = (1+\epsilon)\, u_{ex}$$

In the case of correction by multiplication, we have $\tilde{u} = \tilde{\phi} C$. So for $\tilde{u} = u_{ex}$, we must have

$$\tilde{\phi} C = u_{ex} \quad \Longleftrightarrow \quad (1+\epsilon)\, u_{ex} C = u_{ex}$$

So if the solution does not cancel out on Ω, we must have

$$C = \frac{1}{1+\epsilon} \quad \text{on } \Omega$$

By imposing $C = \frac{1}{1+\epsilon}$ on Γ for FEM instead of $C = 1$, we should get closer to the ϕ-FEM results obtained. We can see in Figure 4.37 and Figure 4.38 that we obtain the expected results for FEM by changing the boundary condition $C = 1$ to $C = \frac{1}{1+\epsilon}$.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
C=1	f = 1	2.78e-02	2.78e-03	2.78e-04	2.78e-05	2.78e-06	2.56e-14
	f = 2	3.72e-02	3.72e-03	3.72e-04	3.72e-05	3.72e-06	2.66e-14
	f = 3	4.66e-02	4.66e-03	4.66e-04	4.66e-05	4.66e-06	2.67e-14
	f = 4	5.42e-02	5.42e-03	5.42e-04	5.42e-05	5.42e-06	3.71e-14
C=1/(1+eps)	f = 1	2.97e-13	2.99e-13	2.98e-13	2.99e-13	2.98e-13	2.97e-13
	f = 2	2.80e-13	2.79e-13	2.79e-13	2.80e-13	2.80e-13	2.80e-13
	f = 3	2.68e-13	2.68e-13	2.68e-13	2.68e-13	2.68e-13	2.68e-13
	f = 4	2.61e-13	2.61e-13	2.61e-13	2.61e-13	2.61e-13	2.61e-13

Figure 4.37: Results by changing FEM boundary conditions on the circle.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
C=1	f = 1	2.78e-02	2.78e-03	2.78e-04	2.78e-05	2.78e-06	2.56e-14
	f = 2	3.72e-02	3.72e-03	3.72e-04	3.72e-05	3.72e-06	2.66e-14
	f = 3	4.66e-02	4.66e-03	4.66e-04	4.66e-05	4.66e-06	2.67e-14
	f = 4	5.42e-02	5.42e-03	5.42e-04	5.42e-05	5.42e-06	3.71e-14
C=1/(1+eps)	f = 1	2.48e-13	2.49e-13	2.48e-13	2.49e-13	2.47e-13	2.48e-13
	f = 2	2.49e-13	2.49e-13	2.49e-13	2.49e-13	2.49e-13	2.49e-13
	f = 3	2.49e-13	2.49e-13	2.49e-13	2.49e-13	2.49e-13	2.49e-13
	f = 4	2.49e-13	2.49e-13	2.49e-13	2.49e-13	2.49e-13	2.49e-13

Figure 4.38: Results by changing FEM boundary conditions on the square.

Remark. *It should be noted, however, that in practice, for example in the case where $\tilde{\phi}$ is a ϕ-FEM solution or an FNO output, this case is not very realistic. There's no reason to expect the form of the perturbation created by the ϕ-FEM solver or by the FNO to be exactly identical to the solution under consideration.*

- **Results on the non-homogeneous case :**

  First, we consider the correction by adding (without IPP) on the Circle problem (Figure 4.39) and on the Square problem (Figure 4.40).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	7.94e-04	7.94e-05	7.94e-06	7.94e-07	7.95e-08	4.13e-11
	f = 3 ; fp = 1	7.93e-04	7.93e-05	7.93e-06	7.94e-07	7.94e-08	6.79e-11
	f = 3 ; fp = 2	6.36e-03	6.36e-04	6.36e-05	6.36e-06	6.36e-07	6.79e-11
	f = 4 ; fp = 1	7.93e-04	7.93e-05	7.93e-06	7.92e-07	7.90e-08	7.32e-10
	f = 4 ; fp = 2	6.35e-03	6.35e-04	6.35e-05	6.35e-06	6.35e-07	7.32e-10
	f = 4 ; fp = 3	2.10e-02	2.10e-03	2.10e-04	2.10e-05	2.10e-06	7.32e-10
f = fp	f = 1	7.95e-04	7.95e-05	7.95e-06	7.96e-07	7.96e-08	1.19e-10
	f = 2	6.36e-03	6.36e-04	6.36e-05	6.36e-06	6.36e-07	4.13e-11
	f = 3	2.10e-02	2.10e-03	2.10e-04	2.10e-05	2.10e-06	6.79e-11
	f = 4	4.77e-02	4.77e-03	4.77e-04	4.77e-05	4.77e-06	7.32e-10
f < fp	f = 1 ; fp = 2	6.37e-03	6.37e-04	6.37e-05	6.37e-06	6.37e-07	1.19e-10
	f = 1 ; fp = 3	2.10e-02	2.10e-03	2.10e-04	2.10e-05	2.10e-06	1.19e-10
	f = 1 ; fp = 4	4.79e-02	4.79e-03	4.79e-04	4.79e-05	4.79e-06	1.19e-10
	f = 2 ; fp = 3	2.10e-02	2.10e-03	2.10e-04	2.10e-05	2.10e-06	4.13e-11
	f = 2 ; fp = 4	4.78e-02	4.78e-03	4.78e-04	4.78e-05	4.78e-06	4.13e-11
	f = 3 ; fp = 4	4.78e-02	4.78e-03	4.78e-04	4.78e-05	4.78e-06	6.79e-11

Figure 4.39: Correction by adding on the Circle with ϕ-FEM in the non-homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.79e-03	1.79e-04	1.79e-05	1.79e-06	1.79e-07	9.83e-12
	f = 3 ; fp = 1	1.79e-03	1.79e-04	1.79e-05	1.79e-06	1.79e-07	1.16e-10
	f = 3 ; fp = 2	1.45e-02	1.45e-03	1.45e-04	1.45e-05	1.45e-06	1.16e-10
	f = 4 ; fp = 1	1.79e-03	1.79e-04	1.79e-05	1.79e-06	1.79e-07	1.13e-09
	f = 4 ; fp = 2	1.45e-02	1.45e-03	1.45e-04	1.45e-05	1.45e-06	1.13e-09
	f = 4 ; fp = 3	3.94e-02	3.94e-03	3.94e-04	3.94e-05	3.94e-06	1.13e-09
f = fp	f = 1	1.79e-03	1.79e-04	1.79e-05	1.79e-06	1.79e-07	1.42e-11
	f = 2	1.45e-02	1.45e-03	1.45e-04	1.45e-05	1.45e-06	9.83e-12
	f = 3	3.94e-02	3.94e-03	3.94e-04	3.94e-05	3.94e-06	1.16e-10
	f = 4	7.79e-02	7.79e-03	7.79e-04	7.79e-05	7.79e-06	1.13e-09
f < fp	f = 1 ; fp = 2	1.45e-02	1.45e-03	1.45e-04	1.45e-05	1.45e-06	1.42e-11
	f = 1 ; fp = 3	3.94e-02	3.94e-03	3.94e-04	3.94e-05	3.94e-06	1.42e-11
	f = 1 ; fp = 4	7.78e-02	7.78e-03	7.78e-04	7.78e-05	7.78e-06	1.42e-11
	f = 2 ; fp = 3	3.94e-02	3.94e-03	3.94e-04	3.94e-05	3.94e-06	9.83e-12
	f = 2 ; fp = 4	7.78e-02	7.78e-03	7.78e-04	7.78e-05	7.78e-06	9.83e-12
	f = 3 ; fp = 4	7.78e-02	7.78e-03	7.78e-04	7.78e-05	7.78e-06	1.16e-10

Figure 4.40: Correction by adding on the Square with ϕ-FEM in the non-homogeneous case.

Then, we consider the correction by multiplying on the Circle problem (Figure 4.41) and on the Square problem (Figure 4.42). We start by considering the same ϕ-FEM scheme as in the homogeneous case, i.e. here we don't impose any boundary conditions.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.16e+00	1.01e-01	9.79e-03	9.76e-04	9.76e-05	2.92e-11
	f = 3 ; fp = 1	9.95e-01	9.71e-02	9.67e-03	9.67e-04	9.67e-05	2.91e-10
	f = 3 ; fp = 2	9.66e-01	9.07e-02	8.85e-03	8.83e-04	8.83e-05	2.91e-10
	f = 4 ; fp = 1	9.45e-01	9.67e-02	9.68e-03	9.68e-04	9.68e-05	1.52e-09
	f = 4 ; fp = 2	9.56e-01	9.25e-02	9.16e-03	9.15e-04	9.15e-05	1.52e-09
	f = 4 ; fp = 3	8.78e-01	8.50e-02	8.33e-03	8.31e-04	8.31e-05	1.52e-09
f = fp	f = 1	6.65e-01	1.01e-01	1.04e-02	1.05e-03	1.05e-04	9.46e-11
	f = 2	5.74e-01	8.67e-02	9.02e-03	9.05e-04	9.05e-05	2.92e-11
	f = 3	5.28e-01	8.22e-02	8.55e-03	8.58e-04	8.59e-05	2.91e-10
	f = 4	5.08e-01	8.05e-02	8.39e-03	8.42e-04	8.42e-05	1.52e-09
f < fp	f = 1 ; fp = 2	5.65e-01	1.96e-01	1.42e-02	1.45e-03	1.46e-04	9.46e-11
	f = 1 ; fp = 3	8.61e-01	2.55e-01	2.45e-02	2.30e-03	2.28e-04	9.46e-11
	f = 1 ; fp = 4	9.44e-01	2.72e+00	3.13e-02	3.46e-03	3.49e-04	9.46e-11
	f = 2 ; fp = 3	4.61e-01	8.23e-02	8.10e-03	8.13e-04	8.14e-05	2.92e-11
	f = 2 ; fp = 4	7.56e-01	1.52e-01	1.49e-02	1.49e-03	1.49e-04	2.92e-11
	f = 3 ; fp = 4	4.54e-01	7.27e-02	7.60e-03	7.63e-04	7.64e-05	2.91e-10

Figure 4.41: Correction by multiplying on the Circle with ϕ-FEM in the non-homogeneous case.

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	6.18e-01	8.29e-02	8.35e-03	8.35e-04	8.35e-05	2.90e-11
	f = 3 ; fp = 1	7.58e-01	9.00e-02	9.01e-03	9.01e-04	9.01e-05	1.78e-10
	f = 3 ; fp = 2	5.60e-01	7.69e-02	7.74e-03	7.74e-04	7.74e-05	1.78e-10
	f = 4 ; fp = 1	8.15e-01	9.28e-02	9.31e-03	9.31e-04	9.31e-05	1.46e-09
	f = 4 ; fp = 2	6.44e-01	8.47e-02	8.51e-03	8.51e-04	8.51e-05	1.46e-09
	f = 4 ; fp = 3	5.61e-01	7.51e-02	7.59e-03	7.59e-04	7.59e-05	1.46e-09
f = fp	f = 1	3.85e-01	6.89e-02	7.51e-03	7.56e-04	7.57e-05	2.02e-11
	f = 2	3.97e-01	7.08e-02	7.58e-03	7.63e-04	7.64e-05	2.90e-11
	f = 3	4.17e-01	7.30e-02	7.77e-03	7.81e-04	7.81e-05	1.78e-10
	f = 4	4.38e-01	7.46e-02	7.98e-03	8.03e-04	8.03e-05	1.46e-09
f < fp	f = 1 ; fp = 2	6.35e-01	1.44e-01	9.08e-03	8.95e-04	8.94e-05	2.02e-11
	f = 1 ; fp = 3	7.85e-01	3.12e-01	1.33e-02	1.56e-03	1.59e-04	2.02e-11
	f = 1 ; fp = 4	8.59e-01	6.36e-01	1.22e-02	1.09e-03	1.09e-04	2.02e-11
	f = 2 ; fp = 3	5.65e-01	6.67e-02	6.65e-03	6.64e-04	6.63e-05	2.90e-11
	f = 2 ; fp = 4	6.56e-01	8.07e-02	8.15e-03	8.13e-04	8.13e-05	2.90e-11
	f = 3 ; fp = 4	5.65e-01	6.86e-02	6.93e-03	6.93e-04	6.93e-05	1.78e-10

Figure 4.42: Correction by multiplying on the Square with ϕ-FEM in the non-homogeneous case.

Observation : We note that the multiplicative corrector using ϕ-FEM seems to succeed, in a similar way to the homogeneous case, to correct the non-homogeneous problem

without imposing the boundary conditions. In fact, there's a subtlety to the scheme we're considering here. Unlike ϕ-FEM (without correction), the scheme is written on $\tilde{\phi}$, which is non-zero at the boundary, and not on ϕ, which is zero at the boundary. This could explain this result, whereas in the case of ϕ-FEM (without correction), we can't avoid imposing boundary conditions.

We will now use the direct method to impose the boundary condition. For this method, we're tempted to consider the solution $\tilde{u} = \tilde{\phi}C + g$ as the solution to the multiplication correction problem. In fact, unlike the classic ϕ-FEM method, the $\tilde{\phi}$ function that replaces our level-set in the formulation is non-zero at the boundary and so, by imposing $C = 1$ at the boundary, we'd have $\tilde{u} = 2g$. To avoid this problem, we will raise the problem by $-g$ and consider $\tilde{u} = (\tilde{\phi} - g)C + g$. We will test this method on the circle (Figure 4.43) and on the square (Figure 4.44).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.26e-03	3.53e-03	1.03e-02	9.81e-04	9.76e-05	4.40e-12
	f = 3 ; fp = 1	1.88e-03	4.01e-02	9.66e-03	9.67e-04	9.67e-05	1.09e-10
	f = 3 ; fp = 2	3.93e-02	7.55e-02	9.47e-03	8.90e-04	8.84e-05	1.09e-10
	f = 4 ; fp = 1	5.67e-03	7.40e-02	9.65e-03	9.68e-04	9.68e-05	9.66e-10
	f = 4 ; fp = 2	6.16e-02	4.13e-02	8.99e-03	9.13e-04	9.14e-05	9.66e-10
	f = 4 ; fp = 3	6.84e-02	6.09e-02	8.87e-03	8.38e-04	8.32e-05	9.66e-10
f = fp	f = 1	1.53e-03	4.47e-05	8.99e-03	1.03e-03	1.04e-04	8.16e-13
	f = 2	5.56e-02	4.21e-02	7.82e-03	8.87e-04	8.96e-05	4.40e-12
	f = 3	5.99e-02	4.10e-02	7.55e-03	8.42e-04	8.50e-05	1.09e-10
	f = 4	6.00e-02	3.96e-02	7.63e-03	8.30e-04	8.39e-05	9.66e-10
f < fp	f = 1 ; fp = 2	3.73e-02	2.84e-02	8.85e-03	1.10e-03	1.12e-04	8.16e-13
	f = 1 ; fp = 3	6.72e-02	5.66e-02	8.43e-03	1.84e-04	1.72e-05	8.16e-13
	f = 1 ; fp = 4	7.76e-02	7.04e-02	4.90e-03	8.90e-04	9.44e-05	8.16e-13
	f = 2 ; fp = 3	2.45e-02	1.98e-02	6.27e-03	7.09e-04	7.09e-05	4.40e-12
	f = 2 ; fp = 4	4.91e-02	4.20e-02	1.13e-02	1.22e-03	1.20e-04	4.40e-12
	f = 3 ; fp = 4	2.80e-02	2.15e-02	6.23e-03	7.05e-04	7.10e-05	1.09e-10

Figure 4.43: Correction by multiplying on the Circle with ϕ-FEM in the non-homogeneous case (direct method).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	2.27e-02	2.03e-02	7.95e-03	8.35e-04	8.36e-05	3.68e-12
	f = 3 ; fp = 1	1.90e-02	1.97e-02	8.81e-03	9.02e-04	9.01e-05	8.07e-11
	f = 3 ; fp = 2	2.18e-02	2.14e-02	6.71e-03	7.75e-04	7.76e-05	8.07e-11
	f = 4 ; fp = 1	1.74e-02	2.27e-02	9.15e-03	9.30e-04	9.31e-05	7.57e-10
	f = 4 ; fp = 2	2.43e-02	2.16e-02	7.66e-03	8.48e-04	8.51e-05	7.57e-10
	f = 4 ; fp = 3	2.25e-02	2.03e-02	6.38e-03	7.55e-04	7.60e-05	7.57e-10
f = fp	f = 1	2.55e-02	2.23e-02	7.34e-03	7.51e-04	7.38e-05	5.88e-13
	f = 2	2.62e-02	2.43e-02	7.01e-03	7.77e-04	7.63e-05	3.68e-12
	f = 3	2.66e-02	2.50e-02	7.07e-03	7.96e-04	7.81e-05	8.07e-11
	f = 4	2.71e-02	2.56e-02	7.19e-03	8.18e-04	8.03e-05	7.57e-10
f < fp	f = 1 ; fp = 2	2.69e-02	2.59e-02	5.71e-03	6.95e-04	7.07e-05	5.88e-13
	f = 1 ; fp = 3	2.93e-02	2.80e-02	5.91e-03	1.33e-04	9.71e-06	5.88e-13
	f = 1 ; fp = 4	3.03e-02	2.93e-02	6.93e-03	4.84e-04	4.61e-05	5.88e-13
	f = 2 ; fp = 3	2.55e-02	2.37e-02	5.00e-03	6.11e-04	6.22e-05	3.68e-12
	f = 2 ; fp = 4	2.81e-02	2.66e-02	6.01e-03	7.61e-04	7.84e-05	3.68e-12
	f = 3 ; fp = 4	2.49e-02	2.32e-02	5.63e-03	6.56e-04	6.64e-05	8.07e-11

Figure 4.44: Correction by multiplying on the Square with ϕ-FEM in the non-homogeneous case (direct method).

We're now going to test imposing boundary conditions with the dual method on the circle (Figure 4.45) and on the square (Figure 4.46).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.41e-04	4.99e-04	8.67e-04	1.32e-03	6.22e-04	6.23e-04
	f = 3 ; fp = 1	6.90e-04	1.21e-03	1.32e-03	2.85e-03	4.56e-03	5.68e-03
	f = 3 ; fp = 2	1.35e-03	1.17e-03	3.46e-03	6.57e-03	7.66e-03	6.92e-03
	f = 4 ; fp = 1	2.33e-04	1.66e-03	2.72e-03	8.51e-04	8.42e-04	8.42e-04
	f = 4 ; fp = 2	4.16e-04	1.39e-02	3.73e-03	1.28e-03	1.26e-03	1.26e-03
	f = 4 ; fp = 3	5.60e-04	2.82e-03	4.14e-03	1.71e-03	1.70e-03	1.70e-03
f = fp	f = 1	2.37e-04	6.44e-04	1.37e-03	5.56e-04	7.84e-04	8.40e-04
	f = 2	2.45e-04	8.77e-04	1.93e-03	1.11e-03	1.20e-03	1.23e-03
	f = 3	2.03e-03	5.82e-03	3.64e-03	9.43e-03	8.64e-03	7.19e-03
	f = 4	6.08e-04	9.74e-03	3.05e-03	2.10e-03	2.09e-03	2.09e-03
f < fp	f = 1 ; fp = 2	4.86e-04	1.39e-03	1.15e-02	7.94e-04	2.79e-03	4.95e-03
	f = 1 ; fp = 3	7.65e-04	8.37e-04	1.12e-03	6.77e-04	7.79e-04	8.18e-04
	f = 1 ; fp = 4	1.06e-03	1.35e-03	2.83e-03	1.14e-03	1.57e-03	1.67e-03
	f = 2 ; fp = 3	2.95e-04	9.93e-04	1.39e-03	1.61e-02	1.15e-03	1.15e-03
	f = 2 ; fp = 4	2.70e-04	1.02e-03	1.18e-03	1.13e-03	1.15e-03	1.15e-03
	f = 3 ; fp = 4	2.67e-03	2.77e-03	2.60e-03	1.25e-02	1.47e-02	2.23e-02

Figure 4.45: Correction by multiplying on the Circle with ϕ-FEM in the non-homogeneous case (dual method).

		eps = 1	eps = 0.1	eps = 0.01	eps = 0.001	eps = 0.0001	eps = 0.0
f > fp	f = 2 ; fp = 1	1.47e-04	6.20e-03	3.95e-03	2.89e-02	5.23e-03	5.14e-03
	f = 3 ; fp = 1	1.22e-04	1.42e-02	1.44e-02	2.48e-03	2.89e-03	3.12e-03
	f = 3 ; fp = 2	1.98e-04	4.01e-02	2.19e-02	6.79e-03	9.32e-03	1.26e-02
	f = 4 ; fp = 1	1.28e-04	1.79e-02	1.47e-02	2.11e-02	1.70e-02	1.58e-02
	f = 4 ; fp = 2	2.38e-04	2.81e-02	2.52e-02	1.59e-02	5.66e-02	3.05e-02
	f = 4 ; fp = 3	2.21e-04	2.78e-02	1.61e-02	3.50e-02	2.42e-01	8.28e-02
f = fp	f = 1	1.13e-04	9.14e-04	1.29e-01	6.95e-04	5.19e-04	5.29e-04
	f = 2	1.94e-04	4.68e-02	1.17e-02	3.04e-01	4.14e-03	3.87e-03
	f = 3	2.19e-04	5.34e-02	7.61e-03	3.26e-03	4.34e-03	6.21e-03
	f = 4	2.58e-04	5.72e-03	2.36e-02	9.99e-02	1.91e-02	1.80e-02
f < fp	f = 1 ; fp = 2	1.94e-04	1.65e-03	2.24e-03	1.10e-03	5.07e-04	5.03e-04
	f = 1 ; fp = 3	2.12e-04	1.18e-03	1.36e-03	3.78e-04	3.81e-04	3.81e-04
	f = 1 ; fp = 4	2.47e-04	1.10e-03	3.14e-03	6.87e-04	6.21e-04	6.19e-04
	f = 2 ; fp = 3	2.08e-04	8.29e-03	5.84e-03	2.10e-03	2.73e-03	2.46e-03
	f = 2 ; fp = 4	2.45e-04	4.52e-02	6.25e-03	1.43e-02	3.07e-03	2.99e-03
	f = 3 ; fp = 4	2.39e-04	1.77e-02	8.37e-03	3.65e-03	4.45e-03	5.05e-03

Figure 4.46: Correction by multiplying on the Square with ϕ-FEM in the non-homogeneous case (dual method).

Observation : It seems that by imposing the boundary conditions with the direct method, the errors are better when ϵ is a bit large, especially for $\epsilon = 1$. For the dual method, it seems also works for imposing boundary conditions. However, we can see that it can become slightly stagnant when ϵ is decreased. It's possible that changing the stabilization parameters could have an impact here.

4.4.2.2 Results on the elevated problem

In this section, we aim to show numerically the interest of elevating the problem. To do this, we will consider the case of the circle with the solution defined in Section 4.1.1.1 and the case of the square with the solution defined in Section 4.1.2.1. We will choose the homogeneous case (i.e. with $\varphi = 0$) with $S = 0.5$ and set $\epsilon = 10^{-3}$.

Results with FEM :

Here, we consider some of the cases considered above, in order to test the correction by multiplying on an elevating problem with FEM (theoretical result presented in Section 4.2.3). We will test this method on the circle (Figure 4.47 and Figure 4.49) and on the square (Figure 4.48 and Figure 4.50) for selected frequencies and by varying m.

		m = 0	m = 1	m = 10	m = 100	m = 1000	m = 10000	Corr_add
f > fp	f = 3 ; fp = 1	1.02e-03	5.59e-05	2.32e-06	2.10e-06	2.10e-06	2.10e-06	2.10e-06
	f = 4 ; fp = 2	9.46e-04	4.16e-05	7.02e-06	6.64e-06	6.62e-06	6.62e-06	6.62e-06
f < fp	f = 1 ; fp = 2	1.06e-04	9.60e-06	6.79e-06	6.64e-06	6.62e-06	6.62e-06	6.62e-06
	f = 1 ; fp = 4	2.17e-04	2.93e-05	2.45e-05	2.42e-05	2.42e-05	2.42e-05	2.42e-05

Figure 4.47: Correction by multiplying on the elevated problem on the Circle with FEM.

		m = 0	m = 1	m = 10	m = 100	m = 1000	m = 10000	Corr_add
f > fp	f = 3 ; fp = 1	9.40e-04	2.54e-05	2.82e-06	2.61e-06	2.61e-06	2.61e-06	2.61e-06
	f = 4 ; fp = 2	8.82e-04	3.47e-05	1.06e-05	1.04e-05	1.04e-05	1.04e-05	1.04e-05
f < fp	f = 1 ; fp = 2	3.98e-05	1.13e-05	1.04e-05	1.04e-05	1.04e-05	1.04e-05	1.04e-05
	f = 1 ; fp = 4	7.39e-05	4.15e-05	4.12e-05	4.12e-05	4.12e-05	4.12e-05	4.12e-05

Figure 4.48: Correction by multiplying on the elevated problem on the Square with FEM.

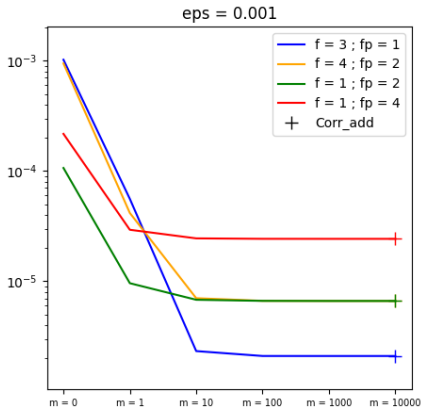

Figure 4.49: Representation of the results on the Circle with FEM.

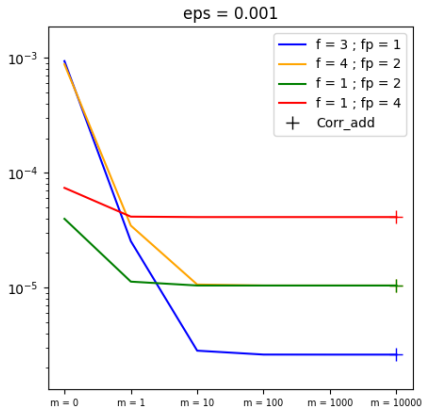

Figure 4.50: Representation of the results on the Square with FEM.

Observation : The numerical results obtained on the circle in Figure 4.47 and on the square 4.48, seem to show that the higher we raise the problem, the better the error. Furthermore, as explained in Section 4.3.2, we can see that by increasing m, the error converges to the error

obtained with the correction by adding (because the solution itself converges to the solution obtained with the correction by adding).

Results with ϕ-FEM :

Now we to test the correction by multiplying on an elevating problem with ϕ-FEM. We will test this method on the circle (Figure 4.51 and Figure 4.53) and on the square (Figure 4.52 and Figure 4.54) for selected frequencies and by varying m. Here, we're using the same scheme as in the homogeneous case, i.e. we're not going to impose the boundary conditions using the direct or dual method.

		m = 0	m = 1	m = 10	m = 100	m = 1000	m = 10000
f > fp	f = 3 ; fp = 1	1.02e-03	7.38e-04	8.91e-03	6.73e-02	6.11e-01	6.03e+00
	f = 4 ; fp = 2	9.45e-04	2.57e-03	9.14e-02	7.45e-02	8.67e-01	8.78e+00
f < fp	f = 1 ; fp = 2	7.76e-06	5.67e-03	6.19e-02	6.93e-01	7.09e+00	7.10e+01
	f = 1 ; fp = 4	8.60e-05	1.06e-02	1.52e-01	2.17e+00	2.31e+01	2.33e+02

Figure 4.51: Correction by multiplying on the elevated problem on the Circle with ϕ-FEM.

		m = 0	m = 1	m = 10	m = 100	m = 1000	m = 10000
f > fp	f = 3 ; fp = 1	9.41e-04	3.49e-03	1.95e-03	6.67e-02	7.82e-01	7.95e+00
	f = 4 ; fp = 2	8.86e-04	1.90e-03	5.71e-03	3.80e-02	3.60e-01	3.58e+00
f < fp	f = 1 ; fp = 2	7.67e-06	1.37e-02	3.07e-02	3.27e-01	3.30e+00	3.30e+01
	f = 1 ; fp = 4	6.38e-05	7.28e-02	3.45e-01	3.77e+00	3.81e+01	3.82e+02

Figure 4.52: Correction by multiplying on the elevated problem on the Square with ϕ-FEM.

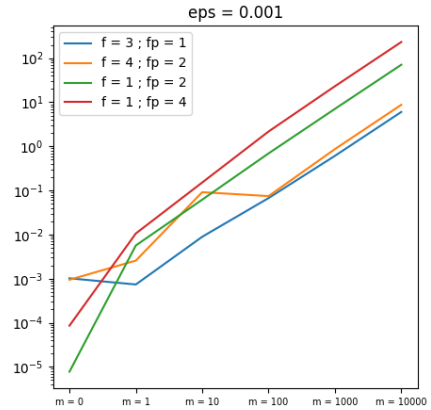

Figure 4.53: Representation of the results on the Circle with ϕ-FEM.

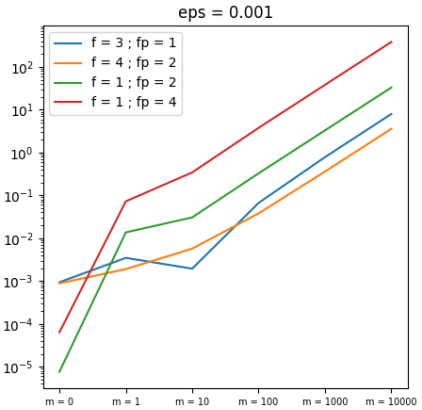

Figure 4.54: Representation of the results on the Square with ϕ-FEM.

Now, we impose the boundary conditions using the dual method, always considering the circle (Figure 4.55 and Figure 4.57) and on the square (Figure 4.56 and Figure 4.58) for selected frequencies and by varying m.

		Corr_mult	m = 0	m = 1	m = 10	m = 100	m = 1000	m = 10000	Corr_add
f > fp	f = 3 ; fp = 1	1.02e-03	1.65e-03	3.27e-04	1.07e-04	5.49e-05	4.94e-05	4.89e-05	8.13e-07
	f = 4 ; fp = 2	9.45e-04	7.66e-04	1.07e-03	5.59e-05	1.12e-04	1.25e-04	1.27e-04	6.51e-06
f < fp	f = 1 ; fp = 2	7.76e-06	6.85e-06	4.56e-04	1.53e-04	1.15e-04	1.11e-04	1.11e-04	6.52e-06
	f = 1 ; fp = 4	8.60e-05	6.70e-05	9.99e-04	4.82e-04	4.16e-04	4.09e-04	4.08e-04	4.90e-05

Figure 4.55: Correction by multiplying on the elevated problem on the Circle with ϕ-FEM (dual method).

		Corr_mult	m = 0	m = 1	m = 10	m = 100	m = 1000	m = 10000	Corr_add
f > fp	f = 3 ; fp = 1	9.41e-04	1.30e-01	1.21e-04	1.19e-04	1.19e-04	1.19e-04	1.19e-04	1.86e-06
	f = 4 ; fp = 2	8.86e-04	9.99e-03	2.09e-04	2.07e-04	2.07e-04	2.07e-04	2.07e-04	1.50e-05
f < fp	f = 1 ; fp = 2	7.67e-06	1.10e-04	2.10e-04	2.07e-04	2.07e-04	2.07e-04	2.07e-04	1.51e-05
	f = 1 ; fp = 4	6.38e-05	2.33e-03	2.61e-04	2.60e-04	2.60e-04	2.60e-04	2.60e-04	8.11e-05

Figure 4.56: Correction by multiplying on the elevated problem on the Square with ϕ-FEM (dual method).

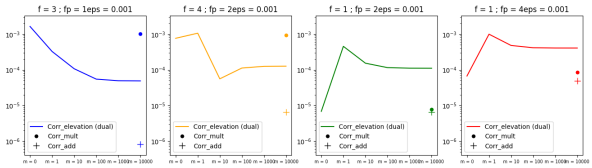

Figure 4.57: Representation of the results on the Circle with ϕ-FEM (dual method).

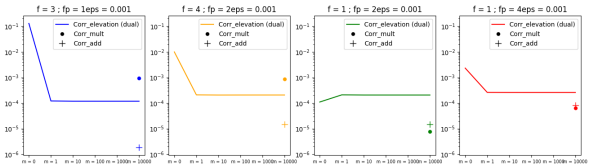

Figure 4.58: Representation of the results on the Square with ϕ-FEM (dual method).

Observation : It would appear that, in the case of multiplication correction on an elevated problem, we are forced to impose the boundary conditions using one of the two methods, unlike multiplication correction without elevation. By imposing boundary conditions using the dual method, it seems that in the case where the frequency of the solution is greater than the frequency of the perturbation (for $f > f_p$), we do reduce the error by increasing m, but it doesn't seem as efficient as in the case with FEM. Indeed, in all the cases considered here, correction by addition gives much better results. Moreover, for $f < f_p$, it would appear that the enhancement is the opposite of the expected effect.

Remark. *Note that the direct method is not applicable in the case of this problem because, as explained in the case of correction without elevation on a non-homogeneous problem, we are in some ways returning to the homogeneous problem. In fact, if we consider*

$$\hat{u} = (\hat{\phi} - g - m)C + (g + m) = (\tilde{\phi} - g)C + (g + m)$$

with $g = 0$ *because we've placed ourselves in the homogeneous case, which amounts to solving the problem without elevation.*

4.4.3 Correction on ϕ-FEM solution

In this section, we will consider the case of the circle with the solution defined in Section 4.1.1.1 with $S = 0.5$, $f = 1$ and $\varphi = 0$ (homogeneous case) and $n_{vert} = 100$.
The aim here is to test the different types of correction presented above on a perturbed solution whose perturbation is not analytically known. We therefore choose to consider

$$\tilde{\phi} = u_{\phi-FEM}$$

with $u_{\phi-FEM}$ the solution $\mathbb{P}_2$ obtained by solving the problem with the ϕ-FEM method.

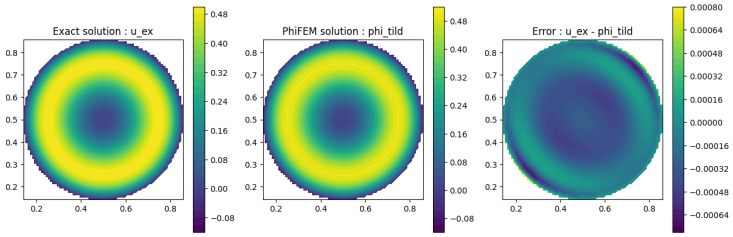

Figure 4.59: Comparison of the exact solution u_{ex} and the ϕ-FEM, $u_{\phi-FEM}$.

We obtain

$$||u_{ex} - u_{\phi-FEM}||_{0,\Omega}^{(rel)} = 8.06e-4.$$

In the following, we will test the different types of correction using the standard FEM method.

Correction by adding :

We begin by testing the addition correction with standard FEM without integration by parts (Figure 4.60), then with integration by parts on the term with $\Delta\tilde{\phi}$ (Figure 4.61).

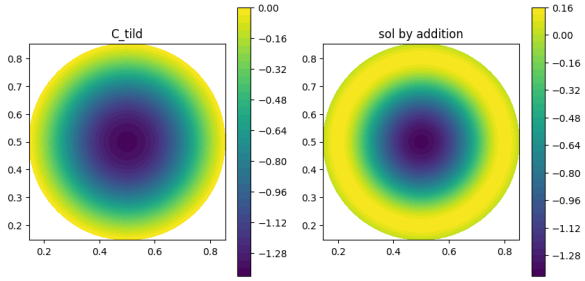

Figure 4.60: Correction of $u_{\phi-FEM}$ by adding with FEM.

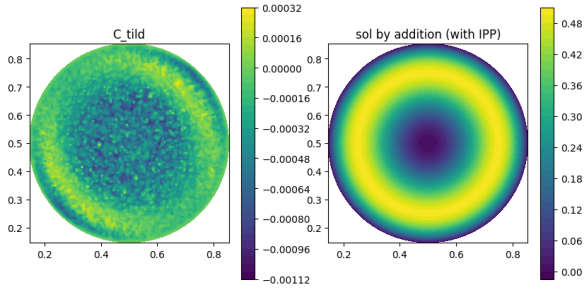

Figure 4.61: Correction of $u_{\phi-FEM}$ by adding (with IPP) with FEM.

We obtain

$$||u_{ex} - u_{add}||_{0,\Omega}^{(rel)} = 2.019 \qquad \text{and } ||u_{ex} - u_{add2}||_{0,\Omega}^{(rel)} = 5.12e-4$$

with u_{add} the solution obtain with the correction by adding and u_{add2} the solution obtained with the correction by adding with integration by parts.

It would therefore seem that correction without integration by parts does not provide the right results at all. This is why, as a first step, we're going to calculate the first derivatives of the solution with FEniCS, according to x and y (Figure 4.62).

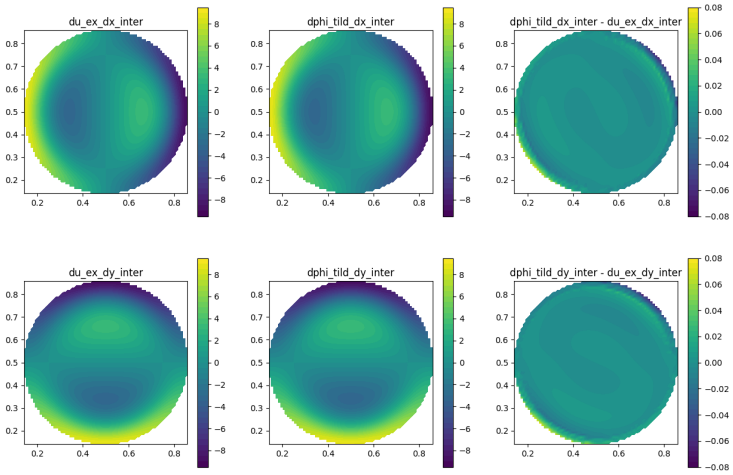

Figure 4.62: First derivatives of the exact solution u_{ex} and the ϕ-FEM solution u_{ϕ}-FEM.

It would appear that there are some errors at the boundary of the domain, but they are fairly

close to the exact derivatives. Let's now calculate the second derivatives of the solution with FEniCS, according to x and y (Figure 4.63).

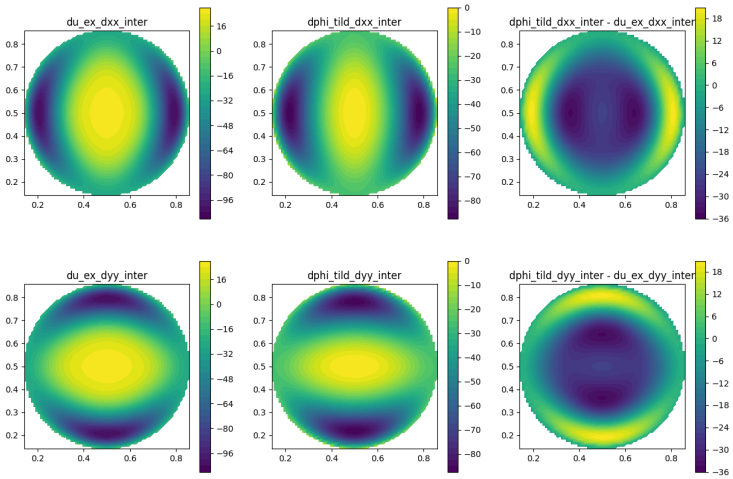

Figure 4.63: Second derivatives of the exact solution u_{ex} and the ϕ-FEM solution u_{ϕ}-FEM.

It seems that the second derivatives are very far from the exact second derivatives, which explains why addition without integration by parts, which uses these second derivatives, doesn't work at all.

Correction by multiplying :

We will now test the correction by multiplying with standard FEM (Figure 4.64) and the correction by multiplying on a elevated problem by taking $m = 1000$ (Figure 4.65).

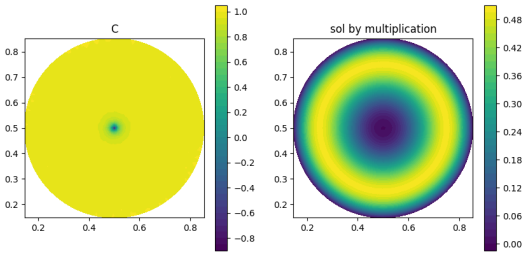

Figure 4.64: Correction of $u_{\phi-FEM}$ by multiplying with FEM.

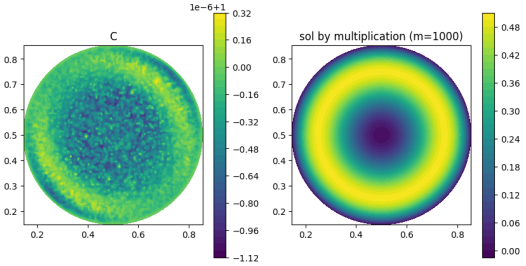

Figure 4.65: Correction of $u_{\phi-FEM}$ by multiplying on an elevated problem with FEM (m=1000).

We obtain

$$||u_{ex} - u_{mult}||_{0,\Omega}^{(rel)} = 2.37e-3 \qquad \text{and } ||u_{ex} - u_{mult,m}||_{0,\Omega}^{(rel)} = 5.12e-4$$

with u_{mult} the solution obtain with the correction by multiplying and $u_{mult,m}$ the solution obtained with the correction by multiplying on an elevated problem.

Remark. *Note that the same error is obtained for correction by multiplying on an elevated problem and correction by addition with integration by parts.*

4.4.4 Correction with FNO

In this section, we will consider the problem presented in Section 4.1.1.2, where we take as geometry the circle and f Gaussian. In practice, this is a very probable case, i.e. one in which no exact solution is known. As explained above, we will take an over-refined solution (calculated with FEM) as our reference solution.
We want to train an FNO to predict the solutions of the problem under consideration and test the correction on its predictions. To do this, we will follow the steps presented in Section 3.4. In the following, we will choose an FNO with 4 Fourier layers.

4.4.4.1 Training the FNO

We start by training the FNO with ϕ-FEM solutions. To do this, we consider a number of training data n_{data} and perform the following steps:

- First, we randomly generate a sample of parameters defined by

$$\left\{\mu_0^{(i)}, \mu_1^{(i)}, \sigma^{(i)}\right\}_{i=1,\dots,n_{data}}$$

with $\sigma \sim \mathcal{U}([0.1, 0.6])$ and $\mu_0, \mu_1 \sim \mathcal{U}([0.5 - \sqrt{2}/4, 0.5 + \sqrt{2}/4])$ with the condition $\phi(\mu_0, \mu_1) < -0.05$.

Remark. *The condition $\phi(\mu_0, \mu_1) < -0.05$ ensures that the center (μ_0, μ_1) of the Gaussian is inside the domain.*

From the parameter sample created, we can generate a Gaussian sample defined by

$$\{f_i\}_{i=1,\dots,n_{data}} = \left\{ \exp\left(-\frac{(x-\mu_0^{(i)})^2 + (y-\mu_1^{(i)})^2}{2(\sigma^{(i)})^2} \right) \right\}_{i=1,\dots,n_{data}}$$

Remark. *As explained in Section 3.4, each Gaussian is in fact evaluated at the nodes of our grid, so they are 2D matrices of size $n_{vert} \times n_{vert}$.*

We can also consider the degrees of freedom $\mathbb{P}_2$, which also form a regular grid, as they are composed of the nodes as well as the midpoints of each segment. So our matrices will be of size $n_{dofs} \times n_{dofs}$ (with $n_{dofs} = 2n_{vert} - 1$). In the following, we will choose n_{vert}=32 and thus $n_{dofs} = 63$.

In our case, the geometry is fixed (the circle with center $(0.5, 0.5)$ and radius $\sqrt{2}/4$), so we will not have a level-set collection in our training data. Similarly, we choose to consider the homogeneous problem and so, since $g = 0$ on Γ, we won't have a collection of Dirichlet conditions in our training data either.

- We can now use the ϕ-FEM scheme associated with the Poisson problem with homogeneous Dirichlet condition defined in Section 2.2.2.2 to solve for each $i = 1, \dots, n_{data}$, the problem

$$\begin{cases} -\Delta(\phi w_i) = f_i, & \text{in } \Omega, \\ u_i = 0, & \text{on } \partial\Omega, \end{cases}$$

with $u_i = \phi w_i$.

We thus have a collection of ϕ-FEM solutions of the problem defined by

$$\{u_i\}_{i=1,\dots,n_{data}} = \{\phi w_i\}_{i=1,\dots,n_{data}}.$$

- From the previous collections, we can now create the X_train and Y_train samples that will enable us to train the FNO. we will start by performing a kind of pre-processing on our data (explained in Section 3.4) by normalizing the source term collection. We thus consider

$$f_{i,norm} = \frac{f_i}{max_{j=1,\dots,n_{data}} ||f_j||_{L^2(\mathcal{O})}}, \quad \forall i \in \{1, \dots, n_{data}\}.$$

We can now define X_train as follows

$$X\_train = \left\{ f_{i,norm}, F^{(x)}_{i,norm}, F^{(y)}_{i,norm}, F^{(xx)}_{i,norm}, F^{(yy)}_{i,norm} \right\}_{i=1,\dots,n_{data}}$$

of size $(n_{data}, n_{dofs}, n_{dofs}, 5)$ where $F^{(x)}_{i,norm}, F^{(y)}_{i,norm}, F^{(xx)}_{i,norm}$ and $F^{(yy)}_{i,norm}$ are respectively the first primitives of $f_{i,norm}$ according to x and y and the second primitives of $f_{i,norm}$ according to x and y. Y_train can also be constructed as follows

$$Y\_train = \{w_i\}_{i=1,\dots,n_{data}}$$

of size $(n_{data}, n_{dofs}, n_{dofs}, 1)$.

Remark. *Considering the term w as training data rather than u ensures that the conditions at the boundary will be accurate at the output of the FNO. Indeed, multiplying the FNO prediction by ϕ guarantees $u = 0$ on Γ.*

Similarly, in the non-homogeneous case, we simply multiply the FNO prediction by ϕ and then add the Dirichlet condition g.

Remark. *In practice, the set defined above is separated into 2 sets: the training set (X_train,Y_train), which trains the FNO, and the validation set (X_val,Y_val), which validates the training. Together, these two sets contain the n_{data} under consideration. In the following, we will consider n_{data} to be the size of our separate training set (after separation : for the generation of 1000 data we have $n_{data} = 875$).*

- We can now train our FNO by minimizing the loss defined by

$$loss_\theta = loss^{(0)}_\theta + loss^{(1)}_\theta + loss^{(2)}_\theta$$

with

$$loss_\theta^{(0)} = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} mse(w_i - w_{\theta,i})$$
$$loss_\theta^{(1)} = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} mse(\nabla_x(w_i) - \nabla_x(w_{\theta,i})) + mse(\nabla_y(w_i) - \nabla_y(w_{\theta,i}))$$
$$loss_\theta^{(2)} = \frac{1}{n_{data}} \sum_{i=1}^{n_{data}} mse(\nabla_{xx}(w_i) - \nabla_{xx}(w_{\theta,i})) + mse(\nabla_{yy}(w_i) - \nabla_{yy}(w_{\theta,i}))$$

where $loss_\theta^{(i)}$ will in practice be called misfit i, $i = 0, 1, 2$.

Remark. *In practice, it may be more interesting to train the FNO directly on ϕw. However, all the results presented here were obtained with loss on w and not ϕw.*

By training our FNO over 4000 epochs with a batch size of 64, we obtain the following misfits as a function of epochs (Figure 4.66):

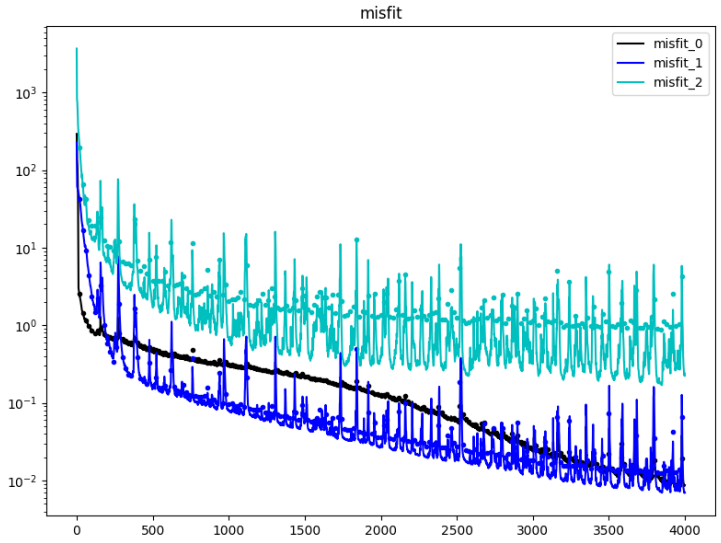

Figure 4.66: Misfits obtained during FNO training by epoch (line - training set, point - validation set).

Results on the validation set :
We're interested here in the $||\cdot||_{0,abs}$ errors obtained at the end of training on the validation set. In fact, we're going to consider different checkpoints in the training, or to be more precise, we're interested in different moments in the training (i.e. we will have 8 similar models whose total number of epochs differs: for the first, we make 500 epochs, for the second we make 1000... up to the last at 4000 epochs). Here are the errors obtained on the validation sample for each of these 8 checkpoints (Figure 4.67):

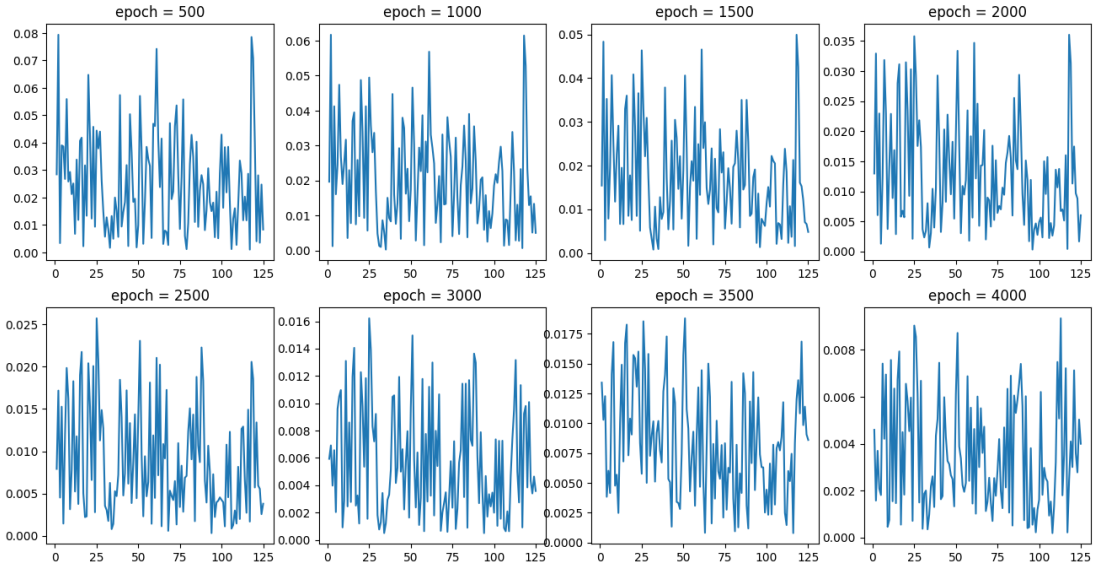

Figure 4.67: Errors obtained on the validation set at different training checkpoints (every 500 epochs).

Here are the mean, standard deviation, minimum and maximum error values obtained on the validation set at these different checkpoints (Figure 4.68), as well as the boxplots of the errors at each checkpoint (Figure 4.69):

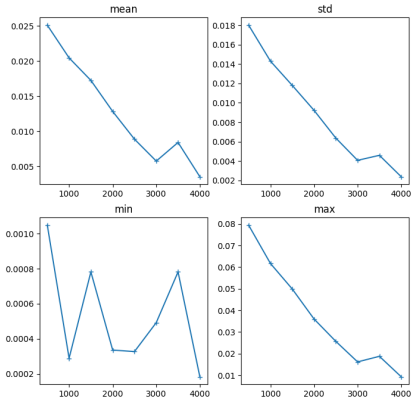

Figure 4.68: Mean, standard deviation, minimum and maximum errors on the validation set according to checkpoints.

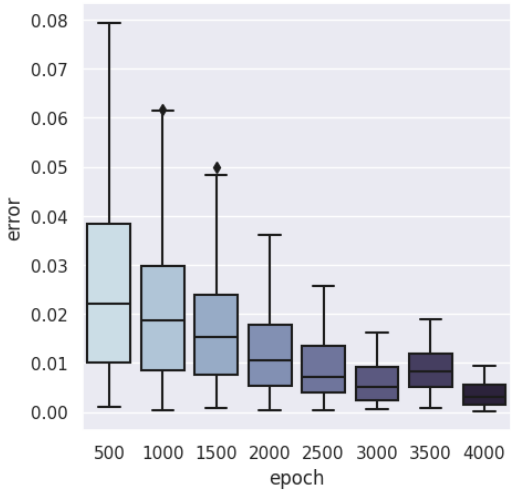

Figure 4.69: Boxplots of the errors on the validation set according to checkpoints.

Observation : It would seem, therefore, that as epochs progress, the errors in the validation sample decrease. In fact, we can see that the mean and standard deviation of the errors decrease according to the epoch. It would therefore seem that training works.

Results on a test set :
This time we're interested in a new test sample of size $n_{test} = 100$, denoted X_test, created in exactly the same way as the training sample (with parameters again created randomly) and we're looking to reproduce exactly the same results as on the validation set. Here are the errors obtained on the test sample for each of these 8 checkpoints (Figure 4.70):

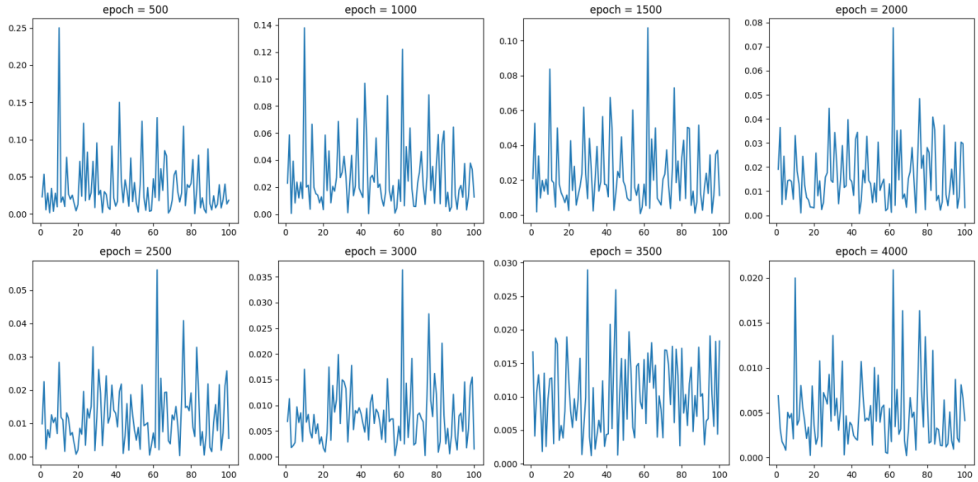

Figure 4.70: Errors obtained on the test set at different training checkpoints (every 500 epochs).

Here are the mean, standard deviation, minimum and maximum error values obtained on the test set at these different checkpoints (Figure 4.71), as well as the boxplots of the errors at each checkpoint (Figure 4.72):

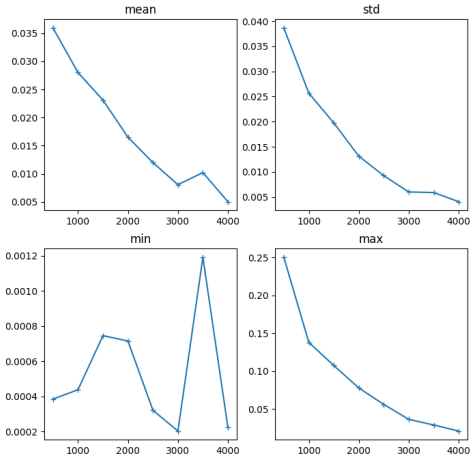

Figure 4.71: Mean, standard deviation, minimum and maximum errors on the test set according to checkpoints.

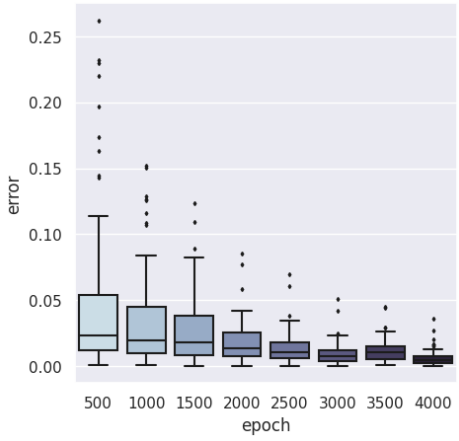

Figure 4.72: Boxplots of the errors on the test set according to checkpoints.

Observation : The same observations can be made as for the validation set.

4.4.4.2 Correction of the FNO prediction

As with the analytical solution and the disturbed solution, the ϕ-FEM method is used to test the various correction methods presented in Section 4.2 on the test sample (of size $n_{test} = 100$) created in Section 4.4.4.1, i.e. correction by addition, correction by multiplication and correction by multiplication on an elevated problem. For each piece of data in the test sample, we consider

$$\tilde{\phi} = u_{FNO} = \phi w_{FNO}$$

with w_{FNO} the prediction made by the FNO on the current test data.

Remark. *Note that, unlike correction on analytic or perturbed solutions, the FNO can only predict the solution at points on the regular grid (i.e. nodes or degrees of freedom $\mathbb{P}^2$). At FNO output, we can therefore only provide our correctors with $\tilde{\phi}$ in $\mathbb{P}_2$.*

For correction by multiplication on a elevated problem, we use the dual method to impose conditions at the boundary.
Here are the errors obtained with the different correction methods, in addition to those obtained directly at the FNO output, according to the checkpoints (Figure 4.73).

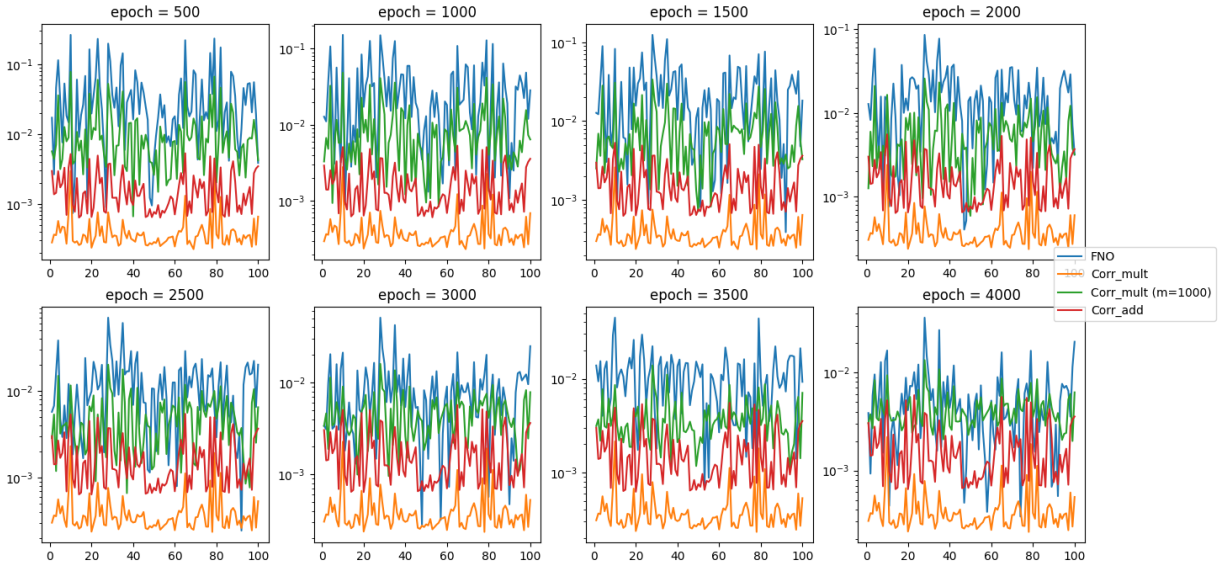

Figure 4.73: Errors obtained with the FNO and with different correction methods according to checkpoints.

We can also plot the error boxplots at each checkpoint (Figure 4.74):

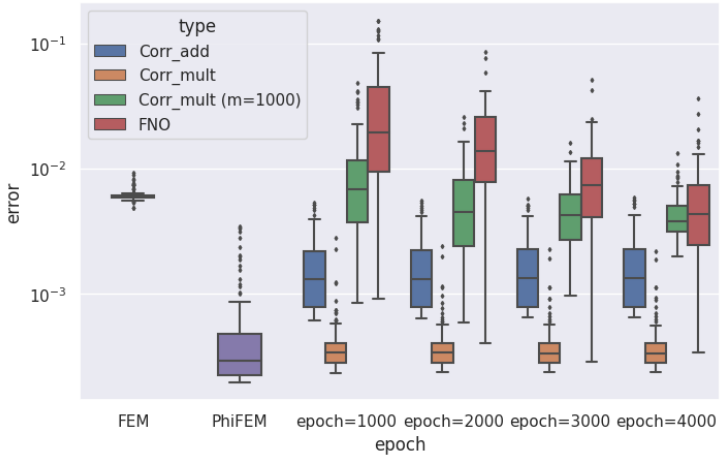

Figure 4.74: Errors obtained with the FNO and with different correction methods according to checkpoints.

Observation : The results here are not really conclusive. Indeed, the 3 correction methods considered did reduce the error made by the FNO, but none seems to be more efficient than ϕ-FEM. In fact, the method that seems to give the best results is correction by multiplication, in contrast to the analytical test cases where the addition method seemed more effective.

4.4.4.3 High degree interpolation

As shown in Section 4.4.4.2, it would seem that considering $\tilde{\phi}$ only in $\mathbb{P}^2$, is not sufficient for the various correction methods applied after the FNO to be more accurate than the initial ϕ-FEM method. For this reason, we're going to attempt to interpolate the solution in order to evaluate this interpolation in a $\mathbb{P}_k$ space of higher degree ($k > 2$). To do this, we will decompose our solution into a series of polynomials, choosing Legendre polynomials.
Explanation :
We want to decompose a function f into a series of Legendre polynomials as follows:

$$f(x,y) = \sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \alpha_{p,q} P_p(x) P_q(y)$$

where the Legendre polynomials are defined for all $n \in \mathbb{N}$ and $x \in \mathbb{R}$ by

$$P_n(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}[(x^2-1)^n]$$

and P and Q are respectively the number of Legendre polynomials associated with x and y. Note that the Legendre polynomials are orthogonal in the space $L^2(]-1,1[)$ and more precisely $\forall n, m \in \mathbb{N}$,

$$\langle P_n, P_m \rangle_{L^2(]-1,1[)} = \int_{-1}^{1} P_n(x) P_m(x) dx = \frac{2}{2n+1} \delta_{nm}.$$

Let us first show that for $p \in \{0,\dots,P-1\}$ and $q \in \{0,\dots,Q-1\}$, the polynomials

$$Q_{p,q}(x,y) = P_p(x)P_q(y)$$

are orthogonal in space $L^2(]-1,1[^2)$:

Remark. *Numerically, we will use the trapezoid method to calculate the scalar product on* $L^2(]-1,1[^2)$.

Let $p, p' \in \{0,\dots,P-1\}$ and $q, q' \in \{0,\dots,Q-1\}$, then by the Legendre polynomials orthogonality , we have

$$\begin{aligned}
\langle Q_{p,q}, Q_{p',q'}\rangle_{L^2(]-1,1[^2)} \int_{-1}^{1}\int_{-1}^{1} Q_{p,q}(x,y)Q_{p',q'}(x,y)\,dxdy &= \int_{-1}^{1}\int_{-1}^{1} P_p(x)P_q(y)P_{p'}(x)P_{q'}(y)\,dxdy \\
&= \int_{-1}^{1} P_p(x)P_{p'}(x)\,dx \times \int_{-1}^{1} P_q(y)P_{q'}(y)\,dy \\
&= \frac{2}{2p+1}\delta_{pp'}\frac{2}{2q+1}\delta_{qq'} \\
&= \frac{4}{(2p+1)(2q+1)}\delta_{(p,q)(p',q')}
\end{aligned}$$

Thus

$$\begin{aligned}
\int_{-1}^{1}\int_{-1}^{1} f(x,y)Q_{p,q}(x,y)\,dxdy &= \langle f, Q_{p,q}\rangle_{L^2(]-1,1[^2)} \\
&= \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} \alpha_{p,q}\langle Q_{p,q}, Q_{p',q'}\rangle_{L^2(]-1,1[^2)} \\
&= \alpha_{p',q'}\langle Q_{p',q'}, Q_{p',q'}\rangle_{L^2(]-1,1[^2)}
\end{aligned}$$

by orthogonality of polynomials $Q_{p,q}$ in $L^2(]-1,1[^2)$.
We deduce

$$\alpha_{p',q'} = \frac{\langle f, Q_{p',q'}\rangle_{L^2(]-1,1[^2)}}{\langle Q_{p',q'}, Q_{p',q'}\rangle_{L^2(]-1,1[^2)}} = \frac{(2p'+1)(2q'+1)}{4}\langle f, Q_{p',q'}\rangle_{L^2(]-1,1[^2)}$$

Remark. *For $x \in [a,b]$, we make a change of variable to bring us back to the interval $[-1,1]$ by considering*

$$\tilde{x} = \frac{2}{b-a}x + \frac{a+b}{a-b}$$

So, assuming that the function f is evaluated on a regular grid, of domain $\mathcal{O}$, of size $N \times N$ (which corresponds to the type of output we get from FNO), then we can calculate the coefficients $\alpha_{p,q}$ for $p \in \{0,\dots,P-1\}$ and $q \in \{0,\dots,Q-1\}$. This gives us an analytical expression

for the function corresponding to a series of Legendre polynomials, enabling us to interpolate our function in all $x, y \in \Omega$.

Decomposition of an analytical function into a Legendre polynomial series :

We want to test Legendre's polynomial series decomposition on the following analytical function

$$f(x,y) = \exp\left(-\frac{(x-\mu_0)^2 + (y-\mu_1)^2}{2\sigma^2}\right)$$

with $x, y \in [0,1]$, $\mu = 0$ and $\sigma = 1$.

Remark. *In practice, with the FNO, it's u that we want to interpolate (for which we don't have an analytical expression) and not f.*

Let's take $P = Q = 5$ and consider the evaluation of f on a regular $N \times N$ grid of $[0,1]^2$ with $N = 100$. After calculating the coefficients $\alpha_{p,q}$ for $p \in \{0,\dots,P-1\}$ and $q \in \{0,\dots,Q-1\}$, we can evaluate the expression

$$f(x,y) = \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} \alpha_{p,q} P_p(x) P_q(y)$$

at any point $x, y \in [0,1]$. Considering, for example, a new regular grid of size $N_2 \times N_2$ of $[0,1]^2$ with $N_2 = 500$, we obtain an absolute error $||\cdot||_{0,\Omega}^{(abs)}$ on Omega between the analytical solution and the expression of the solution in a series of Legendre polynomials of $8.1e-4$ (Figure 4.75).

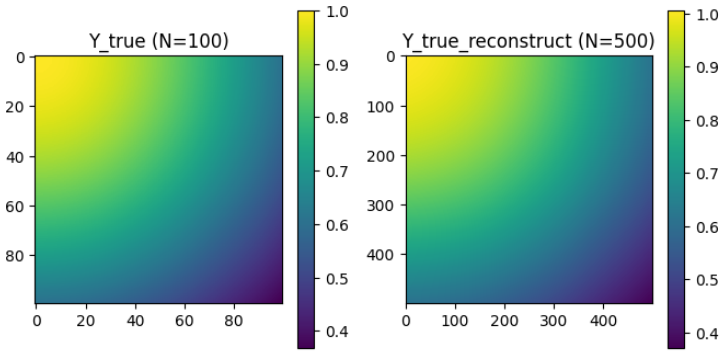

Figure 4.75: Reconstruction of the solution by Legendre polynomials on a new grid of size 500×500.

Decomposition of the FNO predictions into a Legendre polynomial series :

We will again consider the problem presented in Section 4.1.1.2, where we take as geometry the circle and f as being a Gaussian. We again consider the sample X_test (of size $n_{test} = 100$) but this time with $n_{vert} = 300$ (and therefore $n_{dofs} = 599$) to integrate more precisely and thus have a better approximation of the decomposition coefficients. We seek to decompose each FNO output $w_{\theta,i}$, $i = 1,\dots,n_{test}$ into a series of Legendre polynomials, defined by

$$w_{\theta,i}(x,y) = \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} \alpha_{p,q} P_p(x) P_q(y)$$

and thus

$$u_{\theta,i}(x,y) = \phi(x,y)\, w_{\theta,i}(x,y).$$

Remark. *Note that each data in the test sample has its own decomposition.*

In the following, we will consider $P = Q$ and test the decomposition for $P = 4$, $P = 6$ and $P = 8$ on each data of the test sample and at each checkpoint considered. First, we will look at the mean error made by the decomposition into a series of Legendre polynomials, which we will call the mean reconstruction error (Figure 4.76). In other words, for each data item, we calculate the coefficients of the decomposition from the known values of the solution in degrees of freedom $\mathbb{P}_2$, denoted W_pred (of size $(n_{test}, n_{dofs}, n_{dofs})$). We then look at the reconstruction of the solution by the decomposition into a series of Legendre polynomials in these same degrees of freedom $\mathbb{P}_2$, denoted W_pred_reconstruct (of size $(n_{test}, n_{dofs}, n_{dofs})$), then we calculate the error

$$\text{mean\_error\_reconstruction} = ||\text{W\_pred-W\_pred\_reconstruct}||_{0,\mathscr{O}}^{(abs)}$$

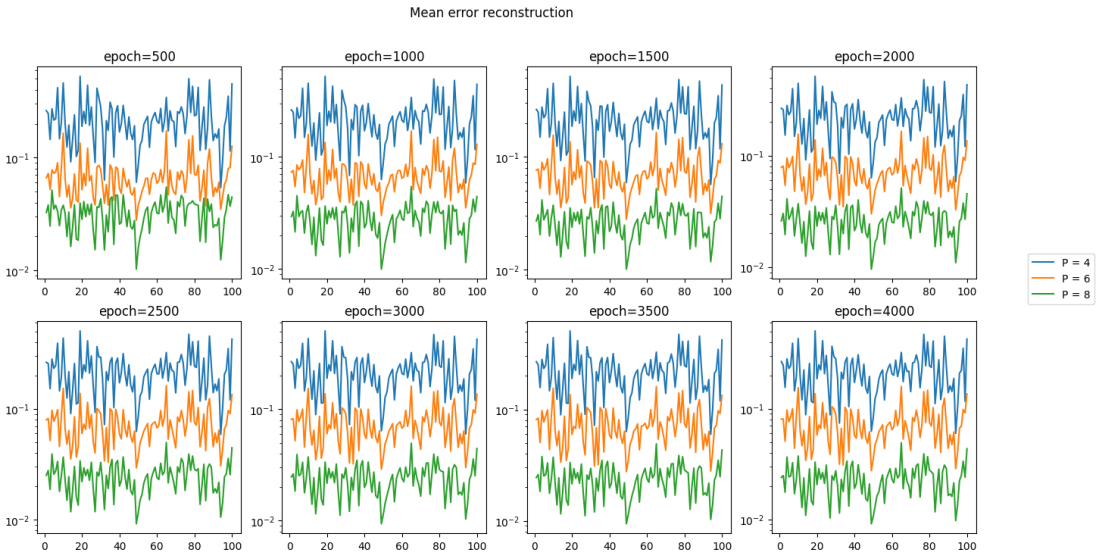

Figure 4.76: Mean reconstruction error for each data in test set (at each checkpoint).

Looking at the results, it seems that the decomposition works. However, it would appear that, on average, we are not as precise as in the analytical case considered with a disturbed solution (Section 4.4.2).
We can now look at the maximum error made by the Legendre polynomial series decomposition, which we will call the maximum reconstruction error (Figure 4.77), ans which is defined by

$$\text{max\_error\_reconstruction} = \max_{\Omega} |\text{W\_pred-W\_pred\_reconstruct}|$$

This will allow us to see if there are any error spikes at certain points.

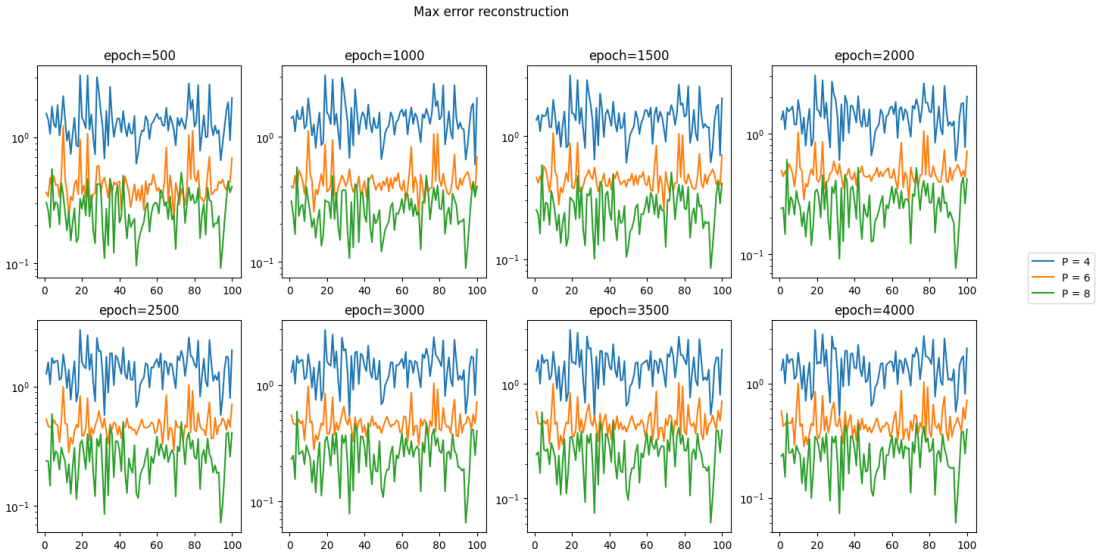

Figure 4.77: Maximal reconstruction error for each data in test set (at each checkpoint).

We can also display solutions in the case of an example (Figure 4.78). we will take the first data item from the first checkpoint to compare W_pred and W_pred_reconstruct.

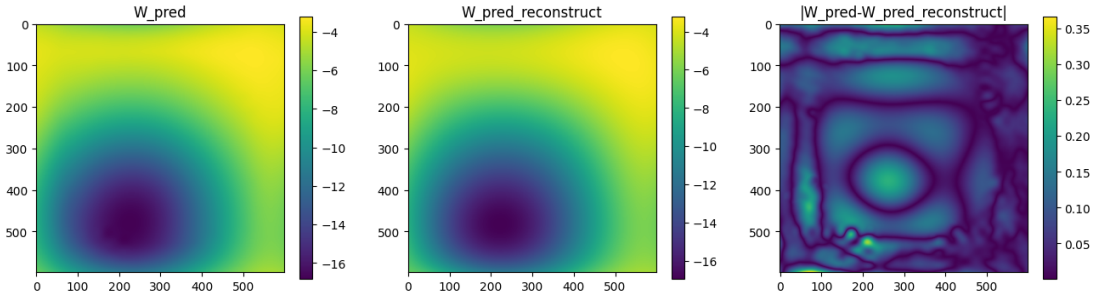

Figure 4.78: Example of result on w (first data from first checkpoint).

It would therefore seem that some regions are more difficult to approach by decomposition than others. We can now look directly at the u solution, rather than w, and consider it on the circle only. To do this, we multiply the predicted solution by ϕ and apply a mask (equal to 1 on the domain and 0 outside). We're then interested in the same errors, but this time only on the solution in our domain. Consider the mean error on the solution (Figure 4.79), defined by

$$\text{mean\_error\_solution} = ||(\text{W\_pred-W\_pred\_reconstruct}) \times \phi||_{0,\Omega}^{(abs)}$$

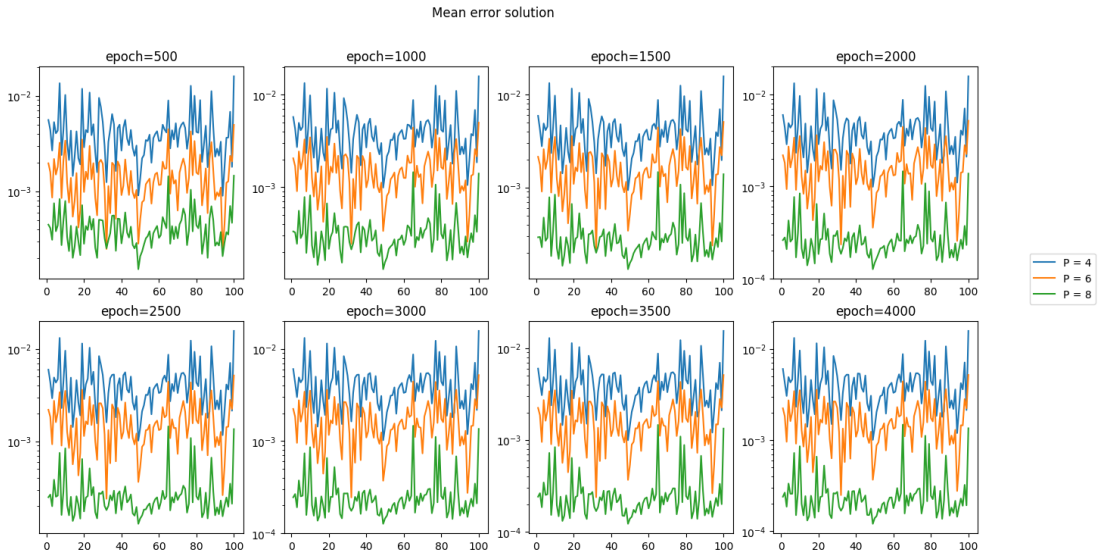

Figure 4.79: Mean solution error for each data in test set (at each checkpoint).

Then we also look at the maximum error on the solution (Figure 4.80), defined by

$$\text{max\_error\_solution} = \max_{\Omega} |(\text{W\_pred-W\_pred\_reconstruct}) \times \phi|$$

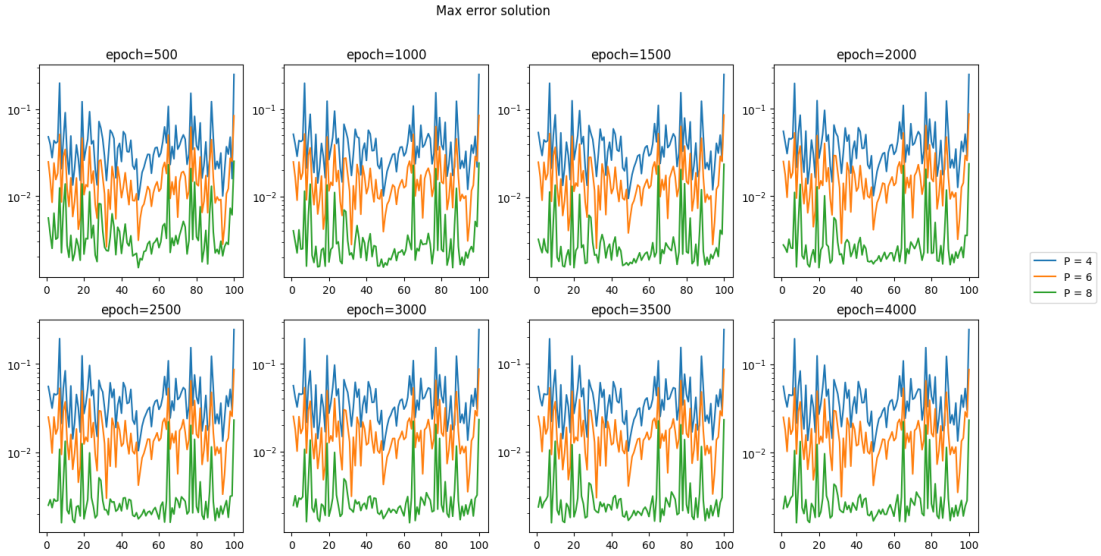

Figure 4.80: Max solution error for each data in test set (at each checkpoint).

We can then compare the solution with the one reconstructed by the series decomposition of Legendre polynomials on the same example (Figure 4.81).

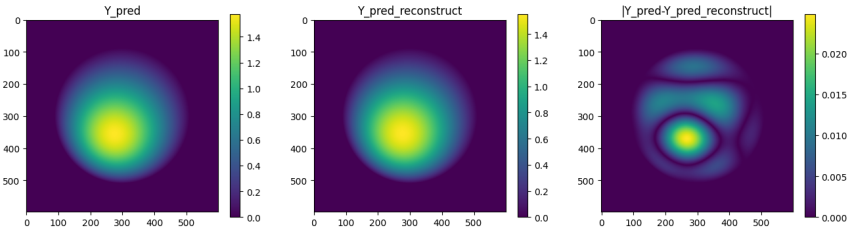

Figure 4.81: Example of result on y (first data from first checkpoint).

We can therefore see that it was more interesting to decompose into a series of Legendre polynomials w and then multiply by ϕ, rather than considering u directly.

Correction with the evaluation of the legendre decomposition :

We have now recovered the $\alpha_{p,q}$ coefficients for each data item in the test sample and at each checkpoint. we will try applying the multiplication correction by taking

$$\tilde{\phi}(x,y) = \left(\sum_{p=0}^{P-1} \sum_{q=0}^{Q-1} \alpha_{p,q} P_p(x) P_q(y) \right) \times \phi(x,y)$$

where x, y are the degrees of freedom associated with $\mathbb{P}^k$ with k large enough.

For each data item at each checkpoint, we will compare the following errors (Figure 4.82): the FNO errors, the errors obtained with the classic multiplication correction (i.e. with $\tilde{\phi}$ in $\mathbb{P}_2$ without Legendre polynomial series decomposition) and finally the errors obtained with the decomposition for $k = 3$ and $k = 5$. To do this, we will simply use the calculated coefficients and evaluate the analytical expression of the decomposition in degrees of freedom $\mathbb{P}_k$ (for $k = 3$ and $k = 5$). Each of these errors will be calculated using the reference solution (over-refined solution obtained with standard FEM).

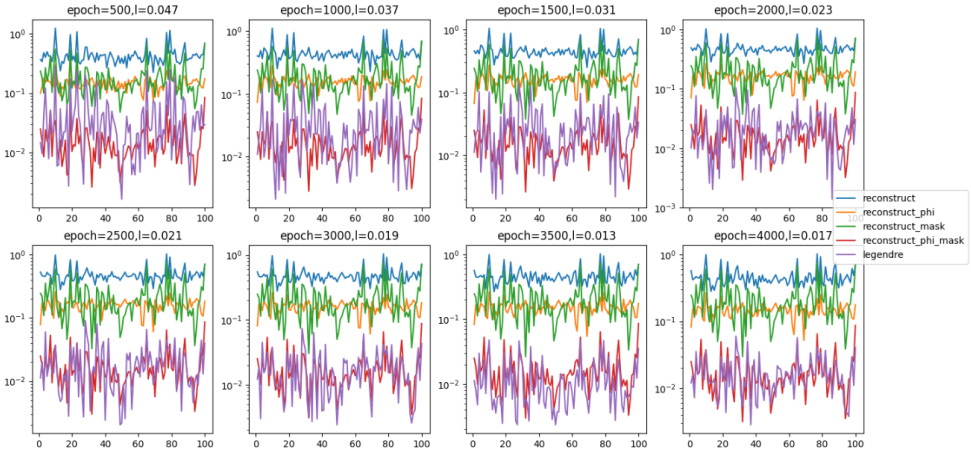

Figure 4.82: Correction by multuiplication with $\tilde{\phi}$ of high degree.

At this stage, the error generated by the decomposition into Legendre polynomial series is probably affecting the correction too much. For this reason, we have not pursued this approach.

4.4.5 Correction with other networks

As explained in Section 3.4 and Section 4.4.4, the general idea of FNO in our application framework is to consider a collection of n_{data} data defined by $\{f_i\}_{i=1,...,n_{data}}$ where each f_i is the evaluation of our source term f associated with parameters i on a regular grid of our domain $\mathcal{O}$ of size $n_{dofs} \times n_{dofs}$ (with n_{dofs} the number of degrees of freedom $\mathbb{P}_1$ or $\mathbb{P}_2$). We then use the FNO to predict the solution to the problem under consideration (in our case, the Poisson problem with Dirichlet condition) on the points of a regular grid (whose resolution may differ from that of the input data).
As we wish to consider a high-degree $\tilde{\phi}$ for the correction solvers, we tried to interpolate the FNO prediction by decomposing it in the form of a series of Legendre polynomials (Section 4.4.4.3) in order to have an analytical expression of the solution at every point of the domain, which was not successful.
We will now consider a new idea, which consists in using a neural network to predict not a collection of solutions, but a single solution at any point in the domain. More precisely, this time we consider our input data to be a collection of 2D points of size n_{pts}, $\{(x_i, y_i)\}_{i=1,...,n_{pts}}$ and we seek to predict the solution $\{u_i\}_{i=1,...,n_{dots}}$ at each of these points with $u_i = \phi(x_i, y_i)w_i$ and $w_i = w_\theta(x_i, y_i)$.
We then test the correction of the solution predicted by this neural network in order to check whether the problem is due to the FNO. If this network produces the expected results, we can add it to the FNO output as a new layer for considering higher-degree solutions.
Two neural networks will be tested here: the first is a Fully-Connected Multi-Layer Perceptron (Fully-Connected MLP) presented in Section 4.4.5.1 and the second is a Physics-Informed Neural Networks (PINNs) presented in Section 4.4.5.2.
In the following, we'll consider the case of the trigonometric solution on the square defined in Section 4.1.2.1 with $S = 0.5$, $f = 1$ and $\varphi = 0$ (so the problem is homogeneous).

Remark. *For MLP and PINNs, we'll be using pytorch codes supplied and implemented by Emmanuel Franck[8] and Victor Michel-Dansac[9].*

4.4.5.1 MLP

Fully-connected Multi Layer Perceptrons are simple dense (i.e. fully-connected) neural networks that can be represented by Figure 4.83.

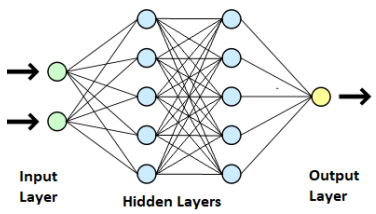

Figure 4.83: Representation of a fully-connected MLP.

[8] Emmanuel Franck : https://irma.math.unistra.fr/~franck/
[9] https://irma.math.unistra.fr/ franck/ : https://irma.math.unistra.fr/~micheldansac/

Consider a regular $\mathbb{P}_2$ grid with $n_{vert} = 64$ nodes in each direction (and therefore $n_{dofs} = 123$ degrees of freedom). Consider the following training sample:

$$X\_train = \{(x_i, y_i)\}_{i=1,\dots,n_{dofs}^2}$$
$$Y\_train = \{u_i\}_{i=1,\dots,n_{dofs}^2}$$

with (x_i, y_i) the coordinates of the degrees of freedom and $u_i = u_{ex}(x_i, y_i)$ the exact solution at each degree of freedom.
The MLP training loss is defined by

$$loss_\theta^{(0)} = \frac{1}{n_{dofs}^2} \sum_{i=1}^{n_{dofs}^2} mse(u_i - \phi(x_i, y_i)\, w_{\theta,i})$$

with $w_{\theta,i}$ the MLP prediction at points (x_i, y_i).
We'll consider a 6-layer network of respective size $\{10, 20, 60, 60, 20, 10\}$ that we'll train over 4000 epochs (Figure 4.84) with a batch size of 16, a learning rate of 0.01 and a hyperbolic tangent activation function.

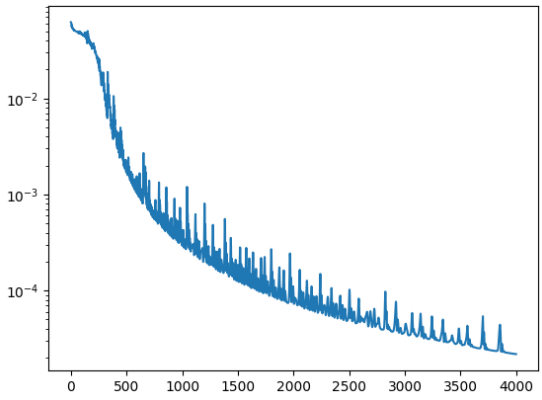

Figure 4.84: Loss obtained during the MLP training.

Taking the training sample to evaluate the model, we obtain an error $||u_{ex} - u_{MLP}||_{0,\Omega}^{(rel)} = 8.09e-3$ and the model seems to learn the solution well (Figure 4.85).

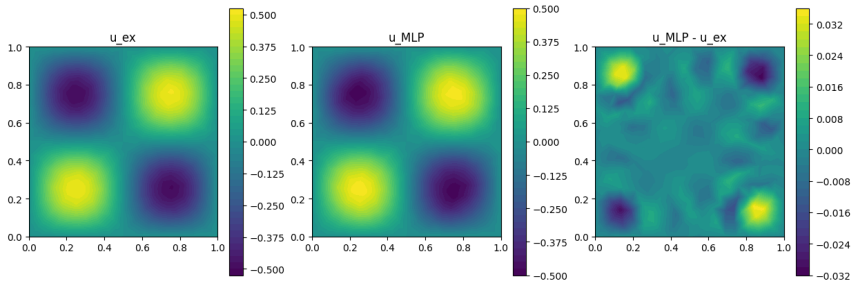

Figure 4.85: Comparaison of the exact solution and the MLP prediciton.

Let's now test the correction by addition without integration by parts (Figure 4.86) with FEM by taking

$$\tilde{\phi} = u_{MLP}$$

with u_{MLP} the MLP prediction.
To do this, we consider $\tilde{\phi}$ in $\mathbb{P}_1 0$ for $n_{vert} = 32$.
Correction with FEM :

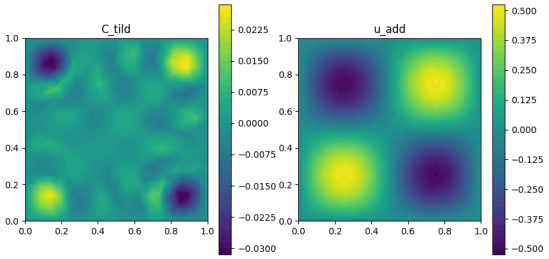

Figure 4.86: Solution obtained with the correction by adding with FEM on the MLP prediction.

We obtain an error $||u_{ex} - u_{MLP}||_{0,\Omega}^{(rel)} = 5.72e-3$. Thus, it would seem that the results are not very promising: for this problem, the FEM error is $2.80e-2$, so we only obtain an error reduction of 4.91.
Derivative of the prediction :
Consequently, we're going to look at the first (Figure 4.87) and second (Figure 4.88) derivatives of the MLP prediction, which will be calculated with FEniCS and compared with the exact derivatives. We obtain the errors $||u_{ex} - u_{MLP}||_{1,\Omega}^{(abs)} = 1.77e-1$ and $||u_{ex} - u_{MLP}||_{2,\Omega}^{(abs)} = 5.136$.

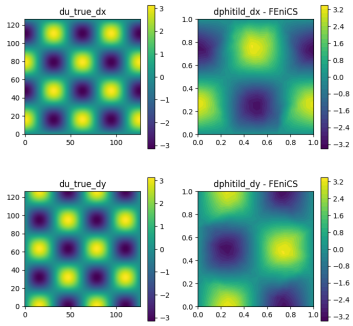

Figure 4.87: First derivatives computed with FEniCS on the MLP prediction.

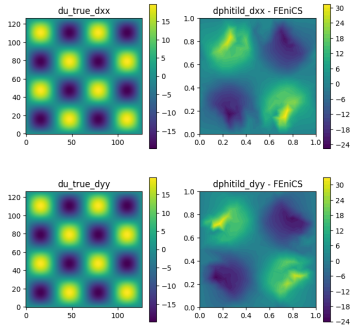

Figure 4.88: Second derivatives computed with FEniCS on the MLP prediction.

Remark. *The derivatives displayed on the left in Figure 4.87 and Figure 4.88 are on the $\mathcal{O}$ domain, so a mask must be applied and the derivatives on the right which are displayed on the Ω domain.*

It seems that the second derivatives are completely wrong, which is not surprising since the model is not trained on the derivatives. To solve this problem, we could very well decide to

add the model's derivatives to the loss as in the FNO loss, but we'll choose here to consider a PINNs whose results are presented in Section 4.4.5.2.

Remark. *Please note that the derivatives of this model cannot be calculated in the same way as for the FNO, i.e. by finite differences but by calculating the derivative of the w_θ model directly.*

4.4.5.2 PINNs

Physics-Informed Neural Networks are neural networks whose structure is variable, but whose loss is the residual of the problem under consideration. We choose to consider an MLP as the model with 6-layer network of respective size $\{10, 20, 60, 60, 20, 10\}$ that we will train over 20000 epochs, a variable learning rate parameter and a hyperbolic sinus activation function. Half of the training is performed with a learning rate of 0.01 and the other half with a learning rate of 0.001.
Here we choose to train the network with n_{pts} points randomly selected in the domain $\mathcal{O}$, and consider for each epoch a new training sample defined by

$$X\_train = \{(x_i, y_i)\}_{i=1,\dots,n_{pts}}$$
$$Y\_train = \{f_i\}_{i=1,\dots,n_{pts}}$$

with (x_i, y_i) the coordinates of the points and $f_i = f(x_i, y_i)$ the second member of the problem evaluated at each points. We will choose $n_{pts} = 20000$.
Considering the Poisson problem, the PINNs loss is defined as the residual of the problem by

$$loss_\theta = \Delta(\phi(x_i, y_i)\, w_{\theta,i}) + f_i$$

with $w_{\theta,i}$ the PINNs prediction at points (x_i, y_i).
At the end of the training, we display the solution, the PINNs prediciton and the difference of the 2 on randomly chosen points in our domain, as well as the loss obtained (i.e. the residual) according to the epochs (Figure 4.89).

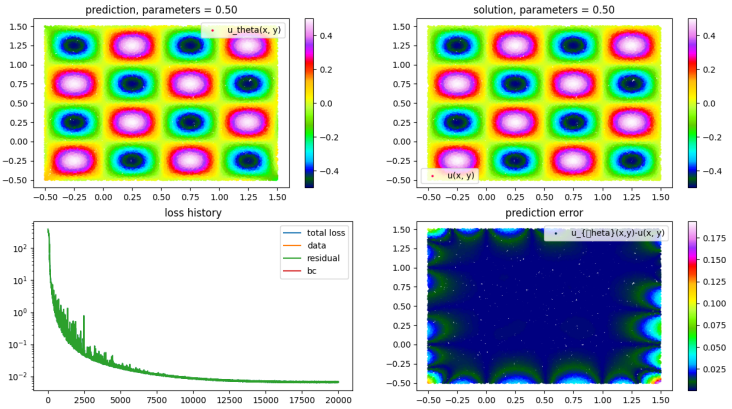

Figure 4.89: PINNs prediction, Exact solution and Loss obtained.

Remark. *It would seem that the prediction is not very good at the border of the $\mathcal{O}$ domain, but as this is a sufficiently large surrounding domain, we won't need the prediction in these areas, far from* Ω.

Correction with FEM :
Let's now test the correction by addition without integration by parts (Figure 4.86) with FEM by taking

$$\tilde{\phi} = u_{PINNS}$$

with u_{PINNS} the PINNS predicition and thus we consider $\tilde{\phi}$ in $\mathbb{P}_{10}$ for $n_{vert} = 32$.
Before the correction, we have the following error.

$$||u_{ex} - \tilde{\phi}||^{(rel)}_{0,\Omega} = 1.93e-3.$$

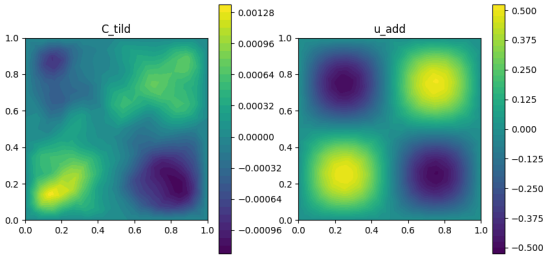

Figure 4.90: Solution obtained with the correction by adding with FEM on the PINNs prediction.

After the correction, we obtain the following error

$$||u_{ex} - \tilde{\phi}C||^{(rel)}_{0,\Omega} = 1.13e-4.$$

Thus, it would seem that the results are very promising: for this problem, the FEM error is

$$||u_{ex} - u_{FEM}||^{(rel)}_{0,\Omega} = 2.80e-2$$

so we obtain an error reduction of 246.60.
Correction with ϕ-FEM :
As the correction on FEM seems to be working, we're going to test the correction by addition without integration by parts (Figure 4.91) with ϕ-FEM.

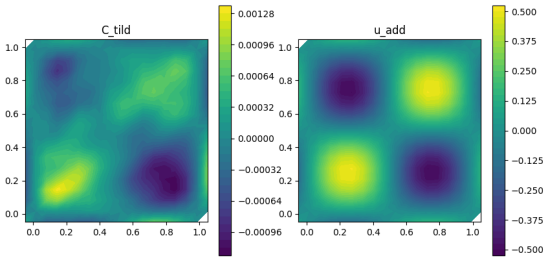

Figure 4.91: Solution obtained with the correction by adding with ϕ-FEM on the PINNs prediction.

After the correction, we obtain the following error

$$||u_{ex} - \tilde{\phi}C||_{0,\Omega}^{(rel)} = 1.27e-4.$$

Thus, it would seem that the results are very promising: for this problem, the FEM error is

$$||u_{ex} - u_{FEM}||_{0,\Omega}^{(rel)} = 1.92e-2$$

so we obtain an error reduction of 151.34.

Derivatives of the prediction :

We're going to look too at the first (Figure 4.92) and second (Figure 4.93) derivatives of the PINNs prediction, which will be calculated with Pytorch and FEniCS and compared with the exact derivatives. We obtain the errors $||u_{ex} - u_{MLP}||_{1,\Omega}^{(abs)} = 6.44e-3$ and $||u_{ex} - u_{MLP}||_{2,\Omega}^{(abs)} = 1.02e-1$.

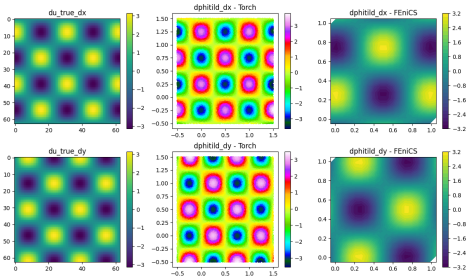

Figure 4.92: First derivatives computed with Pytorch and FEniCS on the PINNs prediction.

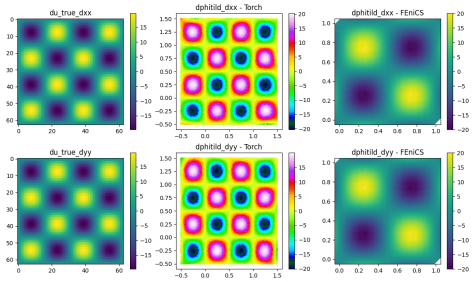

Figure 4.93: Second derivatives computed with Pytorch and FEniCS on the PINNs prediction.

It would therefore seem that the use of PINNs provides results similar to those obtained on perturbed analytic solutions for $\tilde{\phi}$ in $\mathbb{P}_k$ with k large enough ($k = 10$).

5 Conclusion

The main objective of the internship was to combine finite element methods and machine learning in order to solve the Poisson problem with boundary Dirichlet condition. More precisely, we wanted to train an FNO to predict the solutions of a PDE for a given family of problems and apply a correction to them using finite element solvers with the aim of improving the accuracy of the solution.
We have considered 2 types of solution correction methods, the first is an addition correction method and the second is a multiplication correction method, which have been treated by the FEM and ϕ-FEM methods. We have chosen here the case of fairly simple solutions on very simple and fixed geometries, which have enabled us to obtain numerical results on analytical solutions. These test cases have enabled us to confirm that the correction methods considered are functional, and have also confirmed some of the theoretical results obtained. These initial test cases also enabled us to identify some of the limitations of these correction methods, particularly in terms of the degrees of the chosen spaces.
We were then able to apply these various correction methods to the predictions made by the FNO, and it turned out that the results were not satisfactory. As a result, we began to look for methods to increase the degree of the solution in order to bring us back to the results obtained in the analytical cases. The first idea was to decompose the FNO output into a series of polynomials (notably Legendre polynomials) in order to evaluate the solution at any point in our domain and thus consider the high-degree solution.
As the results were inconclusive, we turned to neural networks. Putting the FNO to one side, we set out to build a neural network model capable of predicting a solution at any point in the domain. We began by considering an MLP network, whose lack of learning on derivatives proved problematic. We then moved on to PINNs, which, by learning the solution, also learn its derivatives, enabling us to reduce the error made by conventional methods (FEM and ϕ-FEM) by a factor of around 100 (in the case of correction by addition).
Following on from what has already been done, we could then try adding PINNs to the output of the FNO, in other words, adding a layer output that would replace the decomposition into a series of polynomials and enable us to have the solution at any point in the domain. We could also carry out some documentation work to find more suitable models than the FNO. Finally, we could consider more complex and time-varying geometries (such as 3D organ geometries).

References

[1] Allal Bedlaoui. *Les éléments finis : de la théorie à la pratique*. URL: https://www.academia.edu/36497995/Les_%C3%A9l%C3%A9ments_finis_de_la_th%C3%A9orie_%C3%A0_la_pratique (visited on 08/16/2023).

[2] Allal Bedlaoui. *Les éléments finis : de la théorie à la pratique*. URL: https://www.academia.edu/36497995/Les_%C3%A9l%C3%A9ments_finis_de_la_th%C3%A9orie_%C3%A0_la_pratique (visited on 08/16/2023).

[3] Haim Brezis. *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. New York, NY: Springer New York, 2011. ISBN: 978-0-387-70913-0 978-0-387-70914-7. DOI: 10.1007/978-0-387-70914-7. URL: https://link.springer.com/10.1007/978-0-387-70914-7 (visited on 08/17/2023).

[4] Erik Burman. "Ghost penalty". In: *Comptes Rendus. Mathématique* 348.21 (2010). Number: 21-22, pp. 1217–1220. ISSN: 1778-3569. DOI: 10.1016/j.crma.2010.10.006. URL: http://www.numdam.org/articles/10.1016/j.crma.2010.10.006/ (visited on 07/26/2023).

[5] Erik Burman et al. "CutFEM: Discretizing geometry and partial differential equations". In: *International Journal for Numerical Methods in Engineering* 104.7 (2015). Number: 7 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/nme.4823, pp. 472–501. ISSN: 1097-0207. DOI: 10.1002/nme.4823. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/nme.4823 (visited on 07/26/2023).

[6] Stéphane Cotin et al. *ϕ-FEM: an efficient simulation tool using simple meshes for problems in structure mechanics and heat transfer*.

[7] Michel Duprez, Vanessa Lleras, and Alexei Lozinski. "ϕ-FEM: an optimally convergent and easily implementable immersed boundary method for particulate flows and Stokes equations". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 57.3 (May 2023). Number: 3, pp. 1111–1142. ISSN: 2822-7840, 2804-7214. DOI: 10.1051/m2an/2023010. URL: https://www.esaim-m2an.org/10.1051/m2an/2023010 (visited on 07/26/2023).

[8] Michel Duprez, Vanessa Lleras, and Alexei Lozinski. "A new ϕ-FEM approach for problems with natural boundary conditions". In: *Numerical Methods for Partial Differential Equations* 39.1 (2023). Number: 1 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/num.22878, pp. 281–303. ISSN: 1098-2426. DOI: 10.1002/num.22878. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/num.22878 (visited on 07/26/2023).

[9] Michel Duprez and Alexei Lozinski. "ϕ-FEM: A Finite Element Method on Domains Defined by Level-Sets". In: *SIAM Journal on Numerical Analysis* 58.2 (Jan. 2020). Number: 2 Publisher: Society for Industrial and Applied Mathematics, pp. 1008–1028. ISSN: 0036-1429. DOI: 10.1137/19M1248947. URL: https://epubs.siam.org/doi/10.1137/19M1248947 (visited on 07/26/2023).

[10] Zongyi Li et al. *Neural Operator: Graph Kernel Network for Partial Differential Equations*. Mar. 6, 2020. DOI: 10.48550/arXiv.2003.03485. arXiv: 2003.03485[cs,math,stat]. URL: http://arxiv.org/abs/2003.03485 (visited on 08/17/2023).

[11] Zongyi Li et al. *Fourier Neural Operator for Parametric Partial Differential Equations*. Issue: arXiv:2010.08895. May 16, 2021. DOI: 10.48550/arXiv.2010.08895. arXiv: 2010.08895[cs,math]. URL: http://arxiv.org/abs/2010.08895 (visited on 07/26/2023).

[12] Zongyi Li et al. *Fourier Neural Operator with Learned Deformations for PDEs on General Geometries*. July 11, 2022. DOI: 10.48550/arXiv.2207.05209. arXiv: 2207.05209[cs,math]. URL: http://arxiv.org/abs/2207.05209 (visited on 08/17/2023).

[13] Zongyi Li et al. *Physics-Informed Neural Operator for Learning Partial Differential Equations*. July 29, 2023. DOI: 10.48550/arXiv.2111.03794. arXiv: 2111.03794[cs,math]. URL: http://arxiv.org/abs/2111.03794 (visited on 08/17/2023).

[14] *Méthodes numériques pour les EDP*. URL: https://feelpp.github.io/csmi-edp/#/ (visited on 08/16/2023).

[15] Nicolas Moës and Ted Belytschko. "X-FEM, de nouvelles frontières pour les éléments finis". In: *Revue Européenne des Éléments Finis* 11.2 (Jan. 2002). Number: 2-4, pp. 305–318. ISSN: 1250-6559. DOI: 10.3166/reef.11.305-318. URL: https://www.tandfonline.com/doi/full/10.3166/reef.11.305-318 (visited on 07/26/2023).

[16] Alfio Quarteroni et al. *Méthodes numériques: algorithmes, analyse et applications*. Ed. revue et augmentée. Milan: Springer, 2007. ISBN: 978-88-470-0495-5.